IEEE Access
The journal for rapid open access publishing

IEEE Xplore Digital Library
http://ieeexplore.ieee.org/Xplore

The Video Authentication and Camera Identification Database (Video-ACID): A New Database for Video Forensics

Brian C. Hosler, Xinwei Zhao, Owen Mayer, Chen Chen, James A. Shackleford, Matthew C. Stamm

***Abstract*—Modern technologies have made the capture and sharing of digital video commonplace; the combination of modern smartphones, cloud storage, and social media platforms have enabled video to become a primary source of information for many people and institutions. As a result, it is important to be able to verify the authenticity and source of this information, including identifying the source camera model that captured it. While a variety of forensic techniques have been developed for digital images, less research has been conducted towards the forensic analysis of videos. In part, this is due to a lack of standard digital video databases, which are necessary to develop and evaluate state-of-the-art video forensic algorithms. To address this need, in this paper we present the Video Authentication and Camera IDentification (Video-ACID) database, a large collection of videos specifically collected for the development of camera model identification algorithms. The Video-ACID database contains over 12,000 videos from 46 physical devices representing 36 unique camera models. Videos in this database are hand collected in a diversity of real-world scenarios, are unedited, and have known and trusted provenance. In this paper, we describe the qualities, structure, and collection procedure of Video-ACID, which includes clearly marked videos for evaluating camera model identification algorithms. Finally, we provide baseline camera model identification results on these evaluation videos using a state-of-the-art deep-learning technique. The Video-ACID database is publicly available at misl.ece.drexel.edu/video-acid**

***Index Terms*—Forensics, Multimedia Databases, Benchmark Testing, Video Signal Processing**

I. Introduction

THE capture and spread of digital multimedia has exploded over the last several decades. Higher quality and wildly available cameras, like those in modern smartphones, as well as internet applications, such as social media and cloud storage, have allowed the average person to easily document anything from a family vacation to an academic lecture. However, in some scenarios, such as news reporting, legal proceedings, and national security operations, it is critical to know the source and integrity of a given image or video.

To address these issues, researchers have developed forensic algorithms that verify the authenticity and source of digital content [1], [2], [3], [4]. For example, techniques have been developed to identify the processing history of digital images [5], [6], [7], [8], [9], [10], [11], [12], perform image forgery detection [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], as well to identify an image's source device [25], [26], [27] and source camera model [28], [29], [30], [31], [32], [33], [34]. The development of these algorithms has been significantly aided by the availability of several, high quality forensic databases, such as the Dresden Image Database [35] and the Vision Database [36]

While much of forensics research has focused on images, the increasing importance of video has created a growing need for the development of new video forensic techniques. Currently, researchers have developed forensic algorithms to identify video manipulation and forgery [37], [38], detect frame deletion [39], [40], and identify a videos source device [41], [42], [43] and camera model [44]. While this research provides forensic analysis with important investigative capabilities, the development of video forensic algorithms has proceeded at a much slower pace than image forensic algorithms. One significant reason for this is lack of a widely available database of videos suitable for use in developing and benchmarking forensic algorithms. Though some existing forensic databases contain videos, such as the Vision database, the number of videos in these databases are not sufficiently large to train and evaluate modern data-driven video forensic algorithms. As a result, there is a significant need for a large database of unaltered videos of known provenance that is suitable for use in forensic research.

To fill this gap, we present the Video Authentication and Camera Identification database (Video-ACID). This database is a carefully constructed collection of videos that is purposely made for the development and evaluation of video camera model identification algorithms. While this database is intentionally made with camera model identification techniques in mind, we note that the properties of this database, such as diverse codec parameters, allow this database to be useful for the development and evaluations of many forensic algorithms.

The Video-ACID database contains over 12,000 videos from 46 different devices, totaling 36 represented camera models, and 18 different camera manufacturers. Each device was used to capture on average over 250 different videos. To represent real-world scenarios, these videos were manually captured to depict a diversity of content, lighting conditions, and motion. Videos are unedited, and directly output by the camera for which we have physical access to. Additionally, videos in this dataset have already been used in benchmarking a state-of-the-art video camera model identification [44] system.

The remainder of this paper is structured as follows. In Section II, we discuss the need for a standard database of videos designed for the development of camera model identification techniques, limitations of existing databases, and the desired properties of a video forensics database. In Section III, we detail the creation process, attributes of the videos and

Digital Object Identifier: 10.1109/ACCESS.2019.2922145

camera models, and organization of the Video-ACID database. Finally, in Section IV, evaluate the Video-ACID database by conducting a series of experiments using a state-of-the-art video camera model identification system.

II. Motivation

In the past decade, many algorithms have been developed to determine the origin and integrity of digital images. However, less attention has been paid to videos. One significant reason is that there exists no up-to-date databases that are suitable for developing video forensic analysis techniques. Furthermore, capturing a large amount of videos is expensive and time-consuming. To facilitate the research of video forensics, it is critical to provide researchers a standard video database that has the properties suitable for developing new algorithms.

In this section, we discuss existing relevant databases and their limitations for benchmarking video forensic algorithms. Particularly, we will analyse their shortcomings related to video camera model identification algorithms. Additionally, we outline important properties of a good database for training and benchmarking video forensics algorithms. Although we built this database with camera model identification in mind, we note that many of these properties translate well to other video forensic tasks.

A. Existing Datasets

The Dresden Image Database [35] is a commonly used [45], [46], [47], [48] forensic dataset. This database contains over 14,000 images from 25 different camera models. While the dataset was originally intended for source device identificaion, it has been used to benchmark numerous forensic algorithms, including image source camera model identification [49], [46], image forgery detection [47], and more [48]. However, recent research has shown that features learned by image camera model identification algorithms do not transfer well to video source identification [44]. Since the Dresden Image Database does not contain any videos, it is not useful for developing algorithms that requires feature extraction directly from video frames. However, the flexibility and ubiquity of the Dresden database have motivated us in the construction of our own dataset to adopt similar qualities, such as a large number of camera models, many videos of natural scenery, and a database structure that is easy to interact with.

The recently published VISION dataset [36] contains approximately 300 images and between 10 and 30 videos from each of 35 distinct devices. This dataset also features images and videos which have been uploaded to and downloaded from social media networks such as Facebook and Whatsapp. This dataset was collected for the purpose of device identification based on sensor noise patterns. While the VISION database has been useful for developing a number of forensic algorithms [50], [43], the database does not contain a sufficient number of unique, unedited videos to train state-of-the-art video camera model identification algorithms.

Concurrent to the development of our proposed database, a collection of datasets has been developed called the Multimedia Forensics Challenge (MFC) evaluation datasets [51]. The MFC datasets were built for a number of purposes, including evaluating image and video device (not camera model) identification, forgery detection, and event verification. While these MFC datasets contain many images and videos, none are specifically built with camera model identification in mind, highlighting the need for such a database.

B. Motivating Qualities

We now discuss the properties that make a video database suitable for the development of state-of-the-art algorithms. We considered both the important properties of a video forensic dataset, and the requirements specific to a video camera model identification dataset.

Number of Videos Since many forensic algorithms are trained by first extracting features from a large amount of data, the database should contain a large amount of data points (i.e video clips). Recent forensics algorithms rely increasingly on data driven techniques, such as convolutional neural networks. These neural networks require a large amount of training data to achieve high accuracies [52]. Therefore, a dataset that is suitable for the development of these data driven techniques must have a large number of individual data points.

Number of Camera Models In a real world scenario, a forensic investigator may be tasked with differentiating between a large number of camera models. It is important that a camera model identification algorithm is able to differentiate between many camera models. As a result, it is necessary that a camera model identification database contains data from many different classes i.e. camera models. Additionally, a large number of camera models can help to increase the data variety when used for other forensics tasks.

Diversity of Camera Models A forensic investigator may encounter a breadth and depth of camera models. That is, they may encounter camera models from different manufacturers as well as different camera types, such as camcorders, DSLRs, or cell phones. Additionally, they may need to differentiate between very similar camera models, such as a Samsung Galaxy S5 and Samsung Galaxy S7, which are likely to contain similar forensic traces. Therefore, the database should include a diverse set of camera models.

Known and Trusted Provenance An accurate correspondence between label and data point is critical for a successful classifier [53], [54]. For this reason, it is important that videos are collected by trusted agents who have full control over the capture devices, as opposed to crowd-sourced through the internet. Additionally, for forensic research, it is important to know the history of digital content, including processing history. In the case of camera model identification, it is desirable that videos be unedited and in the format directly output from the camera.

Content diversity It is important that a video forensics data set includes videos captured in many different scenes and environments. This is because multimedia forensic algorithms should be robust to variations in depicted content. That is, a video forensic algorithm should not operate differently when presented with different scenery, lighting environments, motion, etc. To ensure this, training must be performed on videos

captured in a variety of settings, that are ideally representative of all possible encountered scenarios.

Duplicate Devices It is possible, particularly when using deep learning techniques, that a camera model identification algorithm learn device-specific features, in addition to the desired model-specific features. For this reason, it is important that a video database oriented toward camera model identification provides some method by which researchers and algorithmic developers can study the influence of these device-specific features on their algorithms. One way to do this is to capture videos using multiple devices of the same make and model.

Codec Diversity The codec and codec parameters used to compress a video are an important consideration for video forensics techniques [2], [39], [37], [55]. These encoding parameters can affect both the videos perceptual quality, as well as the forensic traces left in the video. It is important for a forensic video dataset to include videos that represent the most popular and modern compression techniques. This includes a diversity of encoding parameters.

In light of this, we provide a brief explanation of modern video coding techniques. Modern video compression techniques take advantage of the temporal redundancy of successive frames. Videos encoded using the H.264 codec are compressed in sets known as Groups of Pictures(GOP's). The first frame of each GOP, known as an I-frame, is coded similar to a JPEG images. The rest of the GOP comprises P and B frames, predicted from temporally local frames. Some cameras further compress frames by sending every other row of the captured video, alternating which rows are sent. This is know as interlaced scanning, as opposed to progressive scanning. There are many other parameters to consider when encoding a video, such as the number of frames per second, and the amount of information that is stored per frame. These parameters are further discussed in Section III.

The above qualities are those we find most important to the usability of a dataset. We considered these when designing our proposed dataset. In the following section, we describe the design of the dataset, and how these qualities were considered and implemented.

III. The Video-ACID Dataset

The Video Authentication and Camera Identification Database(Video-ACID) contains 12,173 total videos from 46 devices, comprising 36 unique camera models. These cameras include a variety of different device categories, manufactures, and models. Specifically, our database contains videos from 19 different smartphones or tablets. We also include videos from 10 point-and-shoot digital cameras, 3 digital camcorders, 2 single-lense reflex cameras, and 2 action cameras. Our dataset contains videos from 18 different camera manufacturers, including Apple, Asus, Canon, Fujifilm, GoPro, Google, Huawei, JVC, Kodak, LG, Motorola, Nikon, Nokia, Olympus, Panasonic, Samsung, Sony, and Yi. For nine of the camera models in the Video-ACID database, videos were collected using two or more physical devices of the same make and model. Table I shows the make and model of each class, as well as some properties of the videos recorded using thoses cameras.

A. Capture Procedure

Videos were captured by hand by a team of researchers who had physical access to each device. Additionally, these videos are unaltered, in the original format directly output by the camera. In order to ensure consistency and avoid biases across the Video-ACID database, the following guidelines were developed and used during data collection.

Device Settings Cameras are often configurable to capture videos with many different parameters including frame size and frame rate. Videos in this dataset are captured by cameras operating at their highest quality setting, usually 1080P at 30 frames per second. Digital zoom can introduce distortions into multimedia content and the associated forensic traces, so during the data collection process, all cameras were left at their default zoom level. Many of these camera models have multiple image sensors on a single device. In these scenarios, we refer to the higher quality rear-facing camera, as opposed to the front facing "selfie" camera.

Duration The videos captured for the Video-ACID dataset are each roughly five seconds or more in duration. This number was chosen in light of many constraints. First, videos must be long enough to exhibit forensically significant behavior, providing a lower bound of several GOP sequences in duration. Second, some forensic algorithms operate on individual frames, as opposed to an entire video. In light of this, we would like to maximize the number of frames available in each video. Third, data collection is an expensive process, and we would like to maximize the number of videos that can be recorded in a given time period. We found that videos of five seconds in duration fit all these constraints.

Content Content diversity is important for many forensic tasks. We collected videos from a variety of different scenes with each camera, including near and far-field focus, indoor and outdoor settings, varied lighting conditions, horizontal and vertical capture, and varied background such as greenery, urban sprawl, snowy landscapes, etc. All videos incorporate some sort of motion or change in scene content, lessening the redundancy of frames within a single video. This motion is typically in the form of panning or rotating the camera, or changing the distance of the camera from the scene. Figure 1 shows still frames of different videos in the Video-ACID database, demonstrating the range and variety of scene content in this database.

B. Camera Capture Properties

Table I summarizes the collected videos and the capture parameters of each device. This table contains information about the capture and compression properties of videos recorded by each camera model such as the resolution, codec profile frame rate mode, and GOP structure.

Resolution The resolution of a video corresponds to the width and height of the video in pixels. Additionally, a suffix of 'I' or 'P' is added to indicate the scan type of the video. A video with an interlaced scan is displayed by updating every

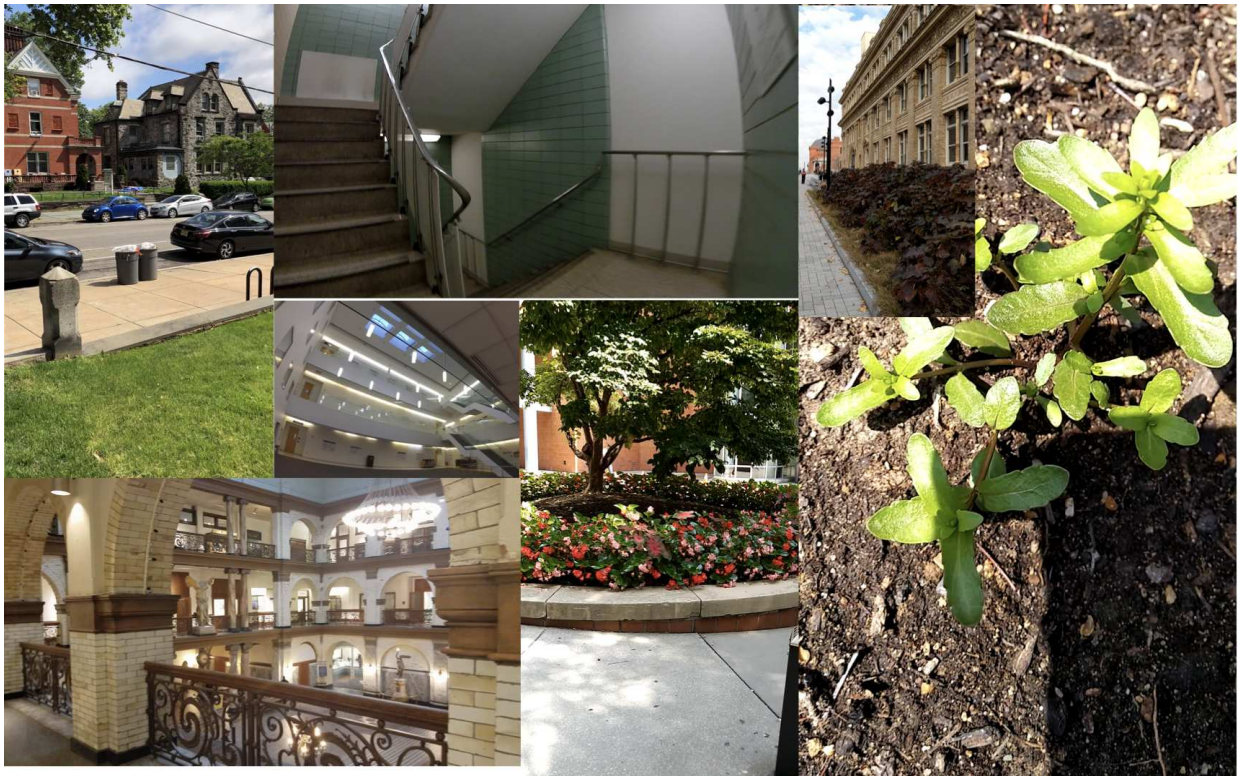


Fig. 1. Sample frames from captured videos.

other row of a frame to reduce the amount of data that needs to be stored. A progressive video is displayed by updating every row of the display for each frame.

Codec Profile Of the 36 camera models used to collect data, most encode captured videos according to the H.264 video coding standard. Associated with this standard are Profiles and Levels, indicating the complexity and speed required to decode the given video. However, two of these cameras encode video using the MJPEG codec, which does not have a profile or level.

The profile of a video determines the complexity needed to decode that video. while the level indicates the speed required. For example ”Baseline” and ”Constrained baseline” videos use Context-Adaptive Variable-Length Coding (CAVLC), while ”Main” and ”High” profile videos use Context-based Adaptive Binary Arethmetic Coding (CABAC). Both encoding schemes are lossless, however CABAC is much more computationally intensive to encode and decode than CAVLC. Notably, B-frames are not available when using the ”Baseline” profiles, but are available when encoding ”Main” and ”High” profile video.

While the Profile indicates a video stream’s complexity in terms of the capability necessary to decode it, a video’s Level indicates bitrate necessary to decode the stream. For example, decoding a 1080p video at 30FPS requires a decoder capable of Level 4 or above. Most modern flagship smartphones use the ”High” profile, while older phones, cheaper phones, and point-and-shoot digital camera are more likely to use the ”Main” or ”Baseline” profiles.

Frame Rate The Video-ACID dataset contains a mix of ”Variable” and ”Fixed” frame rate video. In fixed frame rate videos, each frame is displayed for the same amount of time as every other. In variable frame rate videos, the timing between frames can change. For example, a camera may detect fast motion in a scene, and increase the frame rate to be able to better capture this motion.

GOP structure The length and sequence of a Group of Pictures (GOP) is not fixed by the codec. Instead, as long as a GOP starts with an I-frame, each encoder is allowed to determine its own sequence of P and B frames. A video’s GOP sequence is usually parameterized by the length of of the GOP – the number of frames between I frames – and the maximum number of B frames allowed between anchor frames. In Table I, the N value is the number of frames between I frames, and the M value is the maximum number of B-frames between P-frames For MJPEG-encoded videos there are no predicted frames, so the distance between I-Frames is 1.

As seen in Table I, many cameras will use different M and N parameters. Across similar devices from the same manufacturer however, these parameters are likely to be constant. For example, all the Samsung devices use M=1 and N=30, while Google’s devices use M=1, N=29.

C. Structure

Within Video-ACID, we provide two datasets, a ”Full” dataset, and a ”Duplicate Devices” dataset. The Full dataset contains all videos from all camera models. The Duplicate Devices dataset contains videos from camera models represented by multiple devices. We organize these videos in the following way:

TABLE I
HIGHLIGHTED VIDEO PROPERTIES ASSOCIATED WITH EACH CAMERA MODEL.

ID	Camera Model	Resolution	Codec/Profile	Frame Rate Mode	GOP M	GOP N
M00	Apple iPhone 8 plus	$1920 \times 1080p$	High@4.0	V	2	30
M01	Asus Zenfone 3 Laser	$1920 \times 1080p$	Baseline@4.0	V	1	30
M02	Canon EOS SL1	$1920 \times 1080p$	Constrained Baseline@5.0	F	1	15
M03	Canon EOS T6i	$1920 \times 1080p$	High@4.1	F	3	15
M04	Canon Powershot SX530 HS	$1920 \times 1080p$	Constrained Baseline@4.1	F	1	15
M05	Canon Powershot SX610 HS	$1920 \times 1080p$	Constrained Baseline@4.1	F	1	15
M06	Canon VIXIA HF R800	$1920 \times 1080p$	High@4.0	F	3	15
M07	Fujifilm Finepix S8600	$1280 \times 720p$	MJPEG@-10.1	F	1	1
M08	Fujifilm Finepix XP80	$1920 \times 1080p$	High@4.0	F	3	15
M09	GoPro Hero Session	$1920 \times 1080p$	Main@4.1	F	1	8
M10	Google Pixel 1	$1920 \times 1080p$	High@4.0	V	2	29
M12	Google Pixel 2	$1920 \times 1080p$	High@4.1	V	2	29
M12	Huawei Honor 6X	$1920 \times 1080p$	Constrained Baseline@4.0	V	1	31
M13	Huawei Mate SE	$1280 \times 720p$	Constrained Baseline@3.1	V	1	31
M14	JVC EverioR	$1920 \times 1080i$	High@4.0	V	3	15
M15	Kodak Ektra	$1920 \times 1080p$	High@4.1	V	1	30
M16	LG Q6	$1920 \times 1080p$	High@4.0	V	1	30
M17	LG X Charge	$1920 \times 1080p$	High@4.0	V	1	14
M18	Moto E4	$1920 \times 1080p$	High@4.0	V	1	30
M19	Moto G5 Plus	$1920 \times 1080p$	High@4.0	V	1	30
M20	Nikon Coolpix S33	$1920 \times 1080p$	High@4.0	F	1	30
M21	Nikon Coolpix S3700	$640 \times 480p$	MJPEG@-10.1	F	1	1
M22	Nikon Coolpix S7000	$1920 \times 1080p$	High@4.0	F	1	15
M23	Nokia 6.1	$1920 \times 1080p$	Baseline@4.0	V	1	30
M24	Olympus Stylus Tough TG-860	$1920 \times 1080p$	Constrained Baseline@4.1	F	1	30
M25	Panasonic FZ200	$1920 \times 1080i$	High@4.0	V	3	15
M26	Panasonic HC-V180	$1920 \times 1080i$	High@4.0	V	3	15
M27	Samsung Galaxy J7 Pro	$1920 \times 1080p$	High@4.0	V	1	30
M28	Samsung Galaxy S3	$1920 \times 1080p$	High@4.0	V	1	30
M29	Samsung Galaxy S5	$1920 \times 1080p$	High@4.0	V	1	30
M30	Samsung Galaxy S7	$1920 \times 1080p$	High@4.0	V	1	30
M31	Samsung Galaxy Tab A	$1920 \times 1080p$	High@4.0	V	1	30
M32	Samsung J5-6	$1920 \times 1080p$	High@4.0	V	1	30
M33	Sony Cybershot DSC-WX350	$1920 \times 1080i$	High@4.0	V	2	15
M34	Sony Xperia L1	$1920 \times 1080p$	High@1.0	V	1	30
M35	Yi 4k Action Camera	$3840 \times 2160p$	Main@5.1	F	1	8

Full Dataset The Full dataset contains all videos from all devices in the Video-ACID database. We split this dataset into disjoint sets of training and evaluation videos. To do this, we randomly select 25 videos from each camera model to act as the evaluation set. In the case of multiple devices of the same make and model, these 25 videos are randomly split across the devices. The rest of the videos are left for training.

Many existing camera model identification algorithms operate using patches of an image or video. In light of this, we selected an additional 25 videos from the Nikon Coolpix S3700 because the small frame size limits the number of unique non-overlapping patches that can be extracted. Table II shows the total number of training and evaluation videos for each camera model.

Within the root directory of our "Full" dataset, we have a directory for training data, and another for evaluation data. Within these directories, there is a subdirectory for each camera model. Camera models are identified by both a model number, from 0 to 35, and a name describing the make and model of the camera. Within each of these camera model directories, we separate videos by the device which captured them. For most models, this is just a single subdirectory labeled "DeviceA". When multiple devices of the same model were used to capture videos, there are multiple subdirectories, "DeviceA", "DeviceB", etc.

The videos are named according to the following scheme: MXX_DY_T0000.mp4. MXX is the model number assigned to the camera. DY is the device identifier, e.g. DA for Device A. The prefix of the video number is either 'T' or 'E' indicating whether the video belongs to the training set or evaluation set respectively. Finally, a four-digit number is assigned to index videos captured by the same device. For example, M30_DA_E0010.mp4 refers to the 11th evaluation video captured by device A of a Samsung Galaxy S7. The full filepath is then, "eval/M30_Samsung_Galaxy_S7/DeviceA/M30_DA_E0010.mp4".

Duplicate Devices Dataset The Duplicate Devices dataset contains only those camera models from which videos were captured using multiple devices. This dataset is useful for studying the device dependence of various forensic algorithms. Table III lists the nine camera models and the number of training and evaluation videos in this Duplicate Devices dataset.

From each of these camera models we select the A device and the B device. Videos from the A devices are divided into Train-A and Eval-A, where the Eval-A directory contains 25 videos from each model, and the Train-A directory contains the rest. The videos from the B devices are divided the same way. One camera model, the Google Pixel 1, has videos from three devices. Device C from the Google Pixel 1 is excluded from the Duplicate Devices dataset.

TABLE II
FOLDER NAMES OF NUMBER OF VIDEOS IN THE FULL DATASET.

Folder Name	Devices	Training Videos	Evaluation Videos	Average Duration (s)	Total time (s)
M00_Apple_iPhone_8_plus	1	223	25	5.56	1,380
M01_Asus_Zenfone_3_Laser	1	234	25	6.15	1,592
M02_Canon_EOS_SL1	2	482	25	5.58	2,829
M03_Canon_EOS_T6i	1	202	25	5.39	1,223
M04_Canon_Powershot_SX530_HS	1	220	25	5.98	1,466
M05_Canon_Powershot_SX610_HS	1	191	25	5.34	1,153
M06_Canon_VIXIA_HF_R800	1	226	25	5.62	1,410
M07_Fujifilm_Finepix_S8600	1	178	25	6.37	1,293
M08_Fujifilm_Finepix_XP80	1	212	25	7.17	1,699
M09_GoPro_Hero_Session	1	226	25	5.58	1,400
M10_Google_Pixel_1	3	753	25	6.32	4,919
M11_Google_Pixel_2	1	187	25	5.65	1,197
M12_Huawei_Honor_6X	2	476	25	5.67	2,841
M13_Huawei_Mate_SE	1	498	25	6.36	3,325
M14_JVC_EverioR	1	235	25	6.46	1,681
M15_Kodak_Ektra	2	479	25	5.56	2,804
M16_LG_Q6	2	481	25	6.97	3,528
M17_LG_X_Charge	1	234	25	5.46	1,413
M18_Moto_E4	2	453	25	5.35	2,558
M19_Moto_G5_Plus	1	439	25	5.17	2,398
M20_Nikon_Coolpix_S33	1	196	25	6.17	1,363
M21_Nikon_Coolpix_S3700	1	409	50	7.04	3,230
M22_Nikon_Coolpix_S7000	1	226	25	5.42	1,361
M23_Nokia_6.1	2	476	25	5.49	2,748
M24_Olympus_Stylus_Tough_TG-860	1	221	25	6.11	1,503
M25_Panasonic_FZ200	1	242	25	6.34	1,693
M26_Panasonic_HC-V180	1	240	25	6.60	1,750
M27_Samsung_Galaxy_J7_Pro	2	408	25	6.22	2,694
M28_Samsung_Galaxy_S3	1	230	25	5.82	1,485
M29_Samsung_Galaxy_S5	1	257	25	5.77	1,628
M30_Samsung_Galaxy_S7	1	206	25	5.76	1,330
M31_Samsung_Galaxy_Tab_A	1	232	25	5.46	1,402
M32_Samsung_J5-6	1	203	25	6.40	1,459
M33_Sony_Cybershot_DSC-WX350	1	207	25	6.02	1,396
M34_Sony_Xperia_L1	2	470	25	5.82	2,883
M35_Yi_4k_Action_Camera	1	229	25	5.77	1,465
Total	46	12,173	925	5.96	71,501 s 19:51:41

These directories are structured similarly to the "Full" set, with the different model numbers prefixing the model names. The device subdirectories, because the device is implied, is removed. The videos are directly beneath the camera model directory. For example, train-A/M03_Kodak_Ektra/M03_DA_0010.mp4 is the file path pointing to the 11th training video captured by the A device of the Kodak Ektra.

IV. APPLICATIONS AND BASELINE EVALUATION

In this section, we evaluate the quality of our dataset as a benchmark for forensic algorithms by conducting the following series of experiments. First, we conducted benchmark experiments for camera model identification on our Full dataset. Second, on our Duplicate Dataset, we conducted experiments to investigate device generalization.

A. Camera Model Identification

In our first experiment, we trained and evaluated a state-of-the-art video camera model identification system on our Full dataset. This result establishes a baseline camera model identification accuracy on the Video-ACID database.

To do this, we trained and evaluated a state-of-the-art camera model identification system [44] on the Full dataset. Briefly, this system is a CNN that is trained to output patch-level camera model identification decisions. Furthermore, activations from the CNN are used to fuse neuron activations from multiple patches to render video-level camera model identification decision.

Classifier Training To train the CNN, we used the training videos in our Full dataset to train the classifier according to the procedure in [44]. To extract training patches from the Full dataset, we first start with one camera model and randomly select a training video. We then choose three I-frames from the video also at random. For each of these three frames, we stored every nonoverlapping 256×256 patch, beginning at the top left corner of the frame. This process is repeated until we have extracted 10,000 patches from each class, and then for each camera model for a total of 360,000 patches which are then randomly shuffled.

Our Full dataset primarily contains videos encoded with the H.264 family of codecs (H.264, MPEG-4, etc.). However, the videos in our dataset captured by the Nikon Coolpix S3700 and the Fujifilm Finepix S8600 are encoded using MJPEG. This codec lacks features of the H.264 family of codecs, such

TABLE III
FOLDER NAMES AND NUMBER OF VIDEOS IN THE DUPLICATE DEVICES DATASET.

Folder Name	Train-A	Eval-A	Train-B	Eval-B
M00_Canon_EOS_SL1	237	25	220	25
M01_Google_Pixel_1	235	25	240	25
M02_Huawei_Honor_6X	226	25	225	25
M03_Kodak_Ektra	228	25	226	25
M04_LG_Q6	246	25	210	25
M05_Moto_E4	226	25	202	25
M06_Nokia_6.1	221	25	230	25
M07_Samsung_Galaxy_J7_Pro	228	25	155	25
M08_Sony_Xperia_L1	220	25	225	25

TABLE IV
SINGLE-PATCH AND FUSION ACCURACY OF VIDEO CAMERA MODEL IDENTIFICATION.

Dataset	single-patch accuracy	$F = 3, P = 3$
H.264 only	79.6%	96.9%
H.264 and MJPEG	81.7%	96.0%

TABLE V
SINGLE-PATCH PER-CLASS ACCURACY OF VIDEO CAMERA MODEL IDENTIFICATION.

Camera Model	H.264 Only	H.264 and MJPEG
M00_Apple_iPhone_8_plus	93.9%	85.0%
M01_Asus_Zenfone_3_Laser	76.9%	78.3%
M02_Canon_EOS_SL1	84.2%	92.3%
M03_Canon_EOS_T6i	96.7%	98.0%
M04_Canon_Powershot_SX530_HS	77.5%	74.6%
M05_Canon_Powershot_SX610_HS	89.3%	85.8%
M06_Canon_VIXIA_HF_R800	98.9%	96.7%
M07_Fujifilm_Finepix_S8600		67.7%
M08_Fujifilm_Finepix_XP80	92.2%	96.9%
M09_GoPro_Hero_Session	98.4%	96.6%
M10_Google_Pixel_1	69.1%	79.1%
M11_Google_Pixel_2	85.3%	92.9%
M12_Huawei_Honor_6X	69.8%	73.4%
M13_Huawei_Mate_SE	81.9%	79.5%
M14_JVC_EverioR	95.3%	92.1%
M15_Kodak_Ektra	60.8%	80.0%
M16_LG_Q6	62.4%	71.3%
M17_LG_X_Charge	66.3%	83.9%
M18_Moto_E4	74.6%	62.9%
M19_Moto_G5_Plus	48.8%	54.6%
M20_Nikon_Coolpix_S33	81.1%	93.9%
M21_Nikon_Coolpix_S3700		95.1%
M22_Nikon_Coolpix_S7000	96.5%	95.5%
M23_Nokia_6.1	68.9%	70.3%
M24_Olympus_Stylus_Tough_TG-860	86.5%	95.3%
M25_Panasonic_FZ200	96.7%	93.7%
M26_Panasonic_HC-V180	96.2%	90.8%
M27_Samsung_Galaxy_J7_Pro	71.3%	77.4%
M28_Samsung_Galaxy_S3	75.6%	86.0%
M29_Samsung_Galaxy_S5	53.5%	40.2%
M30_Samsung_Galaxy_S7	64.2%	72.5%
M31_Samsung_Galaxy_Tab_A	67.3%	62.9%
M32_Samsung_J5-6	73.7%	68.9%
M33_Sony_Cybershot_DSC-WX350	89.0%	95.7%
M34_Sony_Xperia_L1	73.4%	61.2%
M35_Yi_4k_Action_Camera	90.8%	98.6%

as predictive coding and variable block sizes. To investigate the effect of the codec on the forensic traces in a video, we train two versions of the classifier. The first classifier is trained on all 36 camera models in our database. The second is trained using only the camera models which produce H.264-encoded video.

Patch Classification First, we evaluate and compare the patch-level camera model identification accuracy of the trained CNNs. To do this, we created an evaluation set by repeating the training patch extraction procedure, however using videos from the evalution set instead. For each camera model, we extracted 1,000 total evaluation patches, yielding a total of 36,000 evaluation patches. We then used the trained classifiers to predict the source camera model of the evaluation patches.

The average single patch classification accuracy for the evaluation set is shown in the second column of Table IV. The CNN trained on the H.264 only dataset, correctly identified the source camera model of 79.6% of evaluation patches.For the CNN trained on both H.264 and MJPEG videos, 81.7% accuracy was achieved.

In Table V we show the average per-class, single-patch accuracy of each trained system. For example, The CNN trained only on H.264 video is able to correctly classify 60.8% of patches taken from the Kodak Ektra. However, the CNN trained on both H.264 and MJPEG videos is able to correctly classify 80.0% of these patches. The accuracies of all camera models range from 50% to 99%.

Table IX shows the confusion matrix obtained using the classifier trained with all 36 camera models. Similarly, Table X shows the confusion matrix obtained using the classifier trained only on H.264-encoded videos.

Video-Level Classification Next, we present camera model identification results on whole videos. To do this, we apply the fusion technique described in [44]. Briefly, the fusion technique fuses neuron activations from P patches from F frames. In this work, we choose $P = 3$, $F = 3$ for a total of 9 patches to fuse.

To evaluate each trained system's fusion accuracy, we randomly selected three I-frames from an evaluation video. Each frame was divided into a set of 256×256 non-overlapping patches, and three patches were randomly selected from each. We use the fusion system in [44] to produce a video-level classification decision. This was repeated for every evaluation video.

The average video-level classification accuracy for the Full dataset is shown in the third column of Table IV. For the H.264 only dataset, 96.9% camera model accuracy was achieved. For the evaluation set comprised of both H.264 and MJPEG

TABLE VI
PER-CLASS FUSION ACCURACY OF VIDEO CAMERA MODEL IDENTIFICATION.

Camera Model	H.264 Only	H.264 and MJPEG
M00_Apple_iPhone_8_plus	100%	100%
M01_Asus_Zenfone_3_Laser	100%	96%
M02_Canon_EOS_SL1	100%	100%
M03_Canon_EOS_T6i	100%	100%
M04_Canon_Powershot_SX530_HS	88%	92%
M05_Canon_Powershot_SX610_HS	100%	100%
M06_Canon_VIXIA_HF_R800	100%	100%
M07_Fujifilm_Finepix_S8600		84%
M08_Fujifilm_Finepix_XP80	100%	100%
M09_GoPro_Hero_Session	100%	100%
M10_Google_Pixel_1	96%	100%
M11_Google_Pixel_2	100%	100%
M12_Huawei_Honor_6X	92%	96%
M13_Huawei_Mate_SE	96%	100%
M14_JVC_EverioR	100%	100%
M15_Kodak_Ektra	84%	100%
M16_LG_Q6	92%	96%
M17_LG_X_Charge	88%	100%
M18_Moto_E4	88%	96%
M19_Moto_G5_Plus	72%	72%
M20_Nikon_Coolpix_S33	100%	100%
M21_Nikon_Coolpix_S3700		98%
M22_Nikon_Coolpix_S7000	100%	100%
M23_Nokia_6.1	100%	100%
M24_Olympus_Stylus_Tough_TG-860	96%	92%
M25_Panasonic_FZ200	100%	100%
M26_Panasonic_HC-V180	100%	100%
M27_Samsung_Galaxy_J7_Pro	100%	96%
M28_Samsung_Galaxy_S3	100%	96%
M29_Samsung_Galaxy_S5	92%	64%
M30_Samsung_Galaxy_S7	96%	92%
M31_Samsung_Galaxy_Tab_A	92%	88%
M32_Samsung_J5-6	96%	96%
M33_Sony_Cybershot_DSC-WX350	100%	100%
M34_Sony_Xperia_L1	92%	76%
M35_Yi_4k_Action_Camera	100%	100%

videos, 96.0% accuracy was achieved. Fusing the multiple patches using either CNN results in a boost in accuracy of over 14% compared to the single-patch classification accuracy. These results are consistent with those reported in [44].

In Table VI we show the per-class video-level classification accuracy of each trained system. Fusing the activations from the CNN trained only on H.264 video results in the correct classification of 84% of patches from the Kodak Ektra. However, when fusing the activations of the CNN trained on both H.264 and MJPEG, 100% accuracy is achieved. The accuracy is between 64% and 100% for all camera models.

B. Device Generalization

In the second experiment, we evaluate the device dependency effects using the Duplicate Devices dataset.

Using the procedure outlined in IV-A, we extracted 10,000 I-frame patches of size 256×256 from each device in the Train-A set. We did the same for devices in the Train-B set. From each device in each evaluation set, we also extracted 1,000 I-frame patches. This resulted in two training sets, each comprising 90,000 patches, and two evaluation sets, each with 9,000 patches. We trained the camera model identification system once on each training dataset, and evaluated it's performance using each of the evaluation datasets.

TABLE VII
SINGLE-PATCH ACCURACY OF CAMERA MODEL IDENTIFICATION SYSTEM WITH VARYING TRAINING AND EVALUATION DEVICES

	Evaluation set A	Evaluation set B
Training set A	82.0%	64.7%
Training set B	74.5%	79.1%

TABLE VIII
$P = F = 3$ FUSION ACCURACY OF CAMERA MODEL IDENTIFICATION SYSTEM WITH VARYING TRAINING AND EVALUATION DEVICES.

	Evaluation set A	Evaluation set B
Training set A	94.7%	81.8%
Training set B	92.0%	95.1%

Table VII shows the average single-patch accuracy of each classifier for each dataset. As shown in Table VII, a CNN trained on only the B devices is able to correctly classify 79.1% of patches from those same devices. That CNN can also correctly classify 74.5% of patches from the A devices. While the CNN trained on only the A devices can correctly classify 82.0% of patches from the A devices, this accuracy falls to 64.7% for patches from the B devices.

To evaluate video-level camera model identification accuracy on this dataset, we employ the fusion system in [44]. For each video in each evaluation set, we randomly selected three I-frames and randomly selected three patches from each of these. We then averaged the accuracy across videos from the A devices, and those from the B devices.

The accuracy of each classifier on each video set is shown in Table VIII. Both CNN's, when evaluating patches from the same devices used during training, achieves close to 95% accuracy. The CNN trained on the B devices correctly classifies 92.0% of patches from the A devices. This is comparable to the CNN's accuracy when classifying patches from B devices. However, The CNN trained on the A devices correctly classifies only 81.8% of patches from the B devices.

When training on the B devices and evaluating on the A device videos with fusion, the accuracy approaches that of training and evaluating on the same device. Interestingly, the system trained on the A devices, when attempting to classify videos from device B, does not achieve the same accuracy.

These results show that when training on one device and evaluating on another device, classification performance decreases relative to training and evaluating on the same device. This suggests that a single device may not be representative of the entire class of camera models. Overfitting to single-devices can affect state-of-the-art video camera model identification systems. The Video-ACID database provides the means for studying this effect.

V. CONCLUSION

In this paper, we proposed a new standard database, VideoACID, that is designed for the study of multimedia forensics on videos. The VideoACID database contains 12,173 video clips of various scenes that were manually captured using 46 unique devices of 36 camera models. The large

amount of video clips ensure the sufficiency and diversity of data for developing and evaluating state-of-art forensic algorithms. Particularly, it satisfies the need for developing source identification algorithms on videos. By conducting a series experiments, we demonstrated the benchmark of the state-of-art forensic video source identification algorithms using the VideoACID database. Moreover, the dataset will grow in both the number of devices and the number of represented camera manufacturers and models, and future work will involve more benchmark evaluations using VideoACID.

VI. Acknowledgments

The authors would like to extend gratitude to Hunter Kippen, Keyur Shah, Shengbang Feng, Belhassen Bayar, and Michael Spanier for their assistance in data collection for this project.

Appendix
Confusion Matricies

TABLE IX
CONFUSION MATRIX OF CAMERA MODEL IDENTIFICATION SYSTEM'S SINGLE-PATCH ACCURACY WHEN TRAINED ON ALL VIDEOACID CAMERA MODELS.

	M00	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35
M00	85.0	-	-	0.5	-	-	-	-	-	-	0.2	-	2.9	3.4	-	3.6	-	0.7	1.1	-	0.1	-	-	-	-	-	-	0.6	1.3	-	-	-	0.2	-	0.4	-
M01	-	78.3	3.2	-	0.1	0.5	-	-	-	-	-	-	0.3	0.9	-	-	13.9	-	1.2	-	-	-	-	-	0.3	-	-	-	-	-	-	1.2	0.1	-	-	-
M02	-	-	92.3	-	0.2	0.4	-	-	-	-	-	-	-	0.4	-	0.2	2.9	-	1.1	-	0.8	-	-	-	0.5	-	-	-	-	-	-	-	-	-	-	1.2
M03	-	-	0.6	98.0	-	-	-	-	-	-	1.1	-	-	0.1	-	-	-	-	-	-	0.1	0.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
M04	-	1.4	9.2	-	74.6	11.3	-	-	-	0.7	-	-	-	0.2	-	-	0.5	-	0.2	-	0.1	-	-	-	1.6	-	-	-	-	-	-	-	-	-	-	0.2
M05	-	0.36	3.93	-	4.76	85.83	-	-	-	-	-	-	0.12	-	-	-	2.14	-	1.31	-	-	-	-	-	1.43	-	-	0.12	-	-	-	-	-	-	-	-
M06	-	0.2	-	2.3	-	-	96.7	-	-	0.1	-	-	-	-	0.2	-	-	-	-	-	-	-	0.4	-	-	-	-	-	-	-	-	-	-	0.1	-	-
M07	-	-	-	0.51	-	-	-	67.69	-	-	-	-	-	-	-	-	-	-	-	-	-	30.0	1.79	-	-	-	-	-	-	-	-	-	-	-	-	-
M08	-	-	-	0.1	-	-	-	-	96.9	-	1.3	-	-	-	-	0.4	-	-	-	0.2	-	-	0.5	-	-	-	-	-	-	-	-	-	0.3	-	0.3	-
M09	0.13	-	0.13	-	0.26	-	-	-	-	96.56	-	0.53	-	0.26	-	-	-	-	0.26	-	-	-	-	0.13	0.79	-	-	0.66	0.26	-	-	-	-	-	-	-
M10	0.2	-	0.2	0.2	-	-	-	-	0.9	-	79.1	1.6	-	0.2	-	4.0	0.7	0.5	0.4	6.1	0.4	-	1.6	-	0.1	-	-	-	0.1	0.7	-	-	1.9	-	1.1	-
M11	-	-	-	-	-	-	-	-	-	-	1.4	92.9	-	-	-	0.2	0.3	0.3	0.3	3.4	-	-	-	0.1	-	-	-	-	0.3	0.2	0.1	-	0.5	-	-	-
M12	0.7	0.3	0.4	-	0.2	-	-	-	-	-	0.1	-	73.4	14.7	-	1.8	0.3	0.5	1.5	-	1.2	-	-	0.4	0.5	0.1	-	2.6	0.5	-	-	0.1	0.3	-	0.3	0.1
M13	1.9	0.71	0.71	1.43	-	-	0.24	-	-	-	-	-	12.38	79.52	-	0.95	-	-	-	-	0.48	-	-	-	-	-	-	0.71	0.71	-	-	-	0.24	-	-	-
M14	-	-	-	-	-	-	-	-	-	-	-	-	0.6	-	92.1	-	-	-	-	-	-	-	-	-	-	1.3	0.1	-	-	-	-	-	-	5.9	-	-
M15	-	-	0.1	-	-	-	-	-	0.1	-	0.9	0.3	2.7	0.4	-	80.0	0.2	7.2	0.7	-	0.6	0.1	-	0.1	-	-	-	0.3	0.9	-	-	-	0.1	-	5.3	-
M16	-	3.9	2.4	-	-	1.1	-	-	-	-	-	0.1	2.6	-	0.1	1.4	71.3	0.3	9.5	0.1	0.2	-	-	0.1	5.0	-	-	0.4	0.2	-	-	1.0	0.1	-	0.2	-
M17	0.54	0.43	-	-	-	-	-	-	-	-	-	0.11	0.11	0.22	-	8.77	0.97	83.87	0.32	0.22	0.43	-	-	-	-	-	0.11	-	0.97	1.73	-	-	0.87	-	0.32	-
M18	-	0.71	-	-	-	-	-	-	-	0.83	0.12	0.24	0.71	-	-	1.55	25.12	0.6	62.86	-	0.6	-	-	0.48	2.86	0.12	-	-	0.36	0.12	-	2.62	0.12	-	-	-
M19	-	-	-	-	-	-	-	-	0.5	-	29.5	6.9	-	-	-	0.1	0.2	-	0.2	54.6	-	-	2.2	-	-	-	-	-	0.2	0.8	-	-	4.7	-	0.1	-
M20	0.1	-	-	-	0.1	0.1	-	-	-	-	-	-	0.2	1.4	-	0.8	-	-	0.1	-	93.9	-	0.3	-	0.4	-	-	1.2	0.8	-	-	-	0.1	-	0.5	-
M21	-	-	-	0.26	-	-	-	1.03	1.28	-	-	-	-	-	-	-	-	-	-	0.26	-	95.13	1.79	-	-	-	-	-	-	-	-	-	0.26	-	-	-
M22	-	-	-	-	-	-	-	-	-	-	3.9	-	-	-	-	0.3	-	-	-	-	0.3	-	95.5	-	-	-	-	-	-	-	-	-	-	-	-	-
M23	0.3	0.5	0.2	-	-	-	-	-	-	4.6	-	0.7	0.6	0.2	0.1	0.6	1.0	-	1.8	-	-	-	-	70.3	2.1	0.4	-	6.8	5.8	-	1.3	1.9	-	-	-	0.8
M24	0.2	-	0.3	-	-	0.4	-	-	-	-	-	-	-	2.5	-	0.1	0.3	-	0.6	-	0.2	-	-	-	95.3	-	-	-	-	-	-	-	-	-	0.1	-
M25	-	0.5	-	-	-	0.1	-	-	-	-	-	-	0.1	-	2.7	-	-	-	0.1	-	-	-	-	-	-	93.7	-	0.4	-	-	-	0.1	-	2.3	-	-
M26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.9	-	-	-	-	-	-	-	-	-	-	0.5	90.8	-	-	-	0.2	-	-	7.6	-	-
M27	1.4	-	-	-	-	-	0.1	-	-	0.2	-	-	2.3	8.6	-	0.2	0.7	0.6	0.4	-	0.3	-	-	2.3	0.8	-	-	77.4	3.0	0.1	0.3	0.8	-	-	-	0.5
M28	0.13	0.13	-	-	-	-	-	-	-	-	0.26	0.4	1.19	0.53	-	3.04	0.26	0.4	0.13	-	1.32	-	-	0.13	2.25	-	-	1.32	85.98	1.32	0.26	-	0.53	-	0.26	0.13
M29	0.6	-	-	0.1	-	-	-	-	1.1	0.5	4.5	9.9	0.1	0.1	-	9.1	0.2	5.4	1.0	0.5	0.6	-	0.4	-	-	-	-	0.5	4.4	40.2	10.5	1.0	7.9	-	1.4	-
M30	-	-	-	-	-	-	-	-	0.2	0.2	1.2	12.3	-	-	0.1	1.5	-	0.1	-	3.0	-	-	-	2.1	-	-	-	0.6	1.3	3.9	72.5	-	0.8	-	0.2	-
M31	-	0.9	0.9	-	-	0.1	-	-	-	4.5	0.1	0.3	1.6	0.1	-	0.9	9.8	0.6	10.2	-	0.1	-	-	1.4	1.5	0.1	0.1	1.3	0.8	0.2	0.5	62.9	0.6	-	-	0.5
M32	-	-	-	-	-	-	-	-	0.11	0.32	6.17	2.49	0.22	-	-	5.09	0.65	0.43	1.08	6.06	0.76	-	0.65	0.11	0.11	-	-	0.22	1.3	4.0	0.54	0.11	68.94	-	0.65	-
M33	-	0.3	-	-	-	-	-	-	-	-	-	-	-	-	0.9	-	-	-	0.1	-	-	-	-	-	-	1.6	1.1	-	-	0.1	-	-	-	95.7	-	0.2
M34	-	0.5	0.2	0.1	-	-	-	-	0.5	-	2.9	-	0.3	0.5	-	24.4	0.1	4.4	0.5	-	1.4	-	-	-	-	-	-	0.1	-	0.8	-	-	2.1	-	61.2	-
M35	-	0.3	-	-	-	-	-	-	-	0.4	-	-	-	-	-	-	-	-	0.2	-	-	-	-	0.4	0.1	-	-	-	-	-	-	-	-	-	-	98.6

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. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 2013.

[4] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, p. 26, 2011.

[5] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, Feb 2005.

[6] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, ser. MM&Sec '08. New York, NY, USA: ACM, 2008, pp. 11–20. [Online]. Available: http://doi.acm.org/10.1145/1411328.1411333

[7] M. Stamm and K. J. R. Liu, "Blind forensics of contrast enhancement in digital images," in *2008 15th IEEE International Conference on Image Processing*, Oct 2008, pp. 3112–3115.

[8] M. Boroumand and J. Fridrich, "Deep learning for detecting processing history of images," *Electronic Imaging*, vol. 2018, no. 7, pp. 213–1–213–9, 2018. [Online]. Available: https://www.ingentaconnect.com/content/ist/ei/2018/00002018/00000007/art00011

[9] B. Bayar and M. C. Stamm, "Towards order of processing operations detection in jpeg-compressed images with convolutional neural networks," in *Media Watermarking, Security, and Forensics*, 2018.

[10] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, "Cnn-based detection of generic contrast adjustment with jpeg post-processing," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 3803–3807.

[11] M. Barni, E. Nowroozi, and B. Tondi, "Improving the security of image manipulation detection through one-and-a-half-class multiple classification," 2019.

[12] T. Bianchi and A. Piva, "Detection of nonaligned double jpeg compression based on integer periodicity maps," *IEEE transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 842–848, 2012.

[13] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A sift-based forensic method for copymove attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.

[14] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *in Proceedings of Digital Forensic Research Workshop*. Citeseer, 2003.

[15] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 1053–1056.

[16] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 857–867, Dec 2010.

[17] O. Mayer and M. C. Stamm, "Accurate and efficient image forgery detection using lateral chromatic aberration," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1762–1777, July 2018.

[18] 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.

[19] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of jpeg artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

[20] M. Chen, J. Fridrich, J. Lukáš, and M. Goljan, "Imaging sensor noise as digital x-ray for revealing forgeries," in *International Workshop on Information Hiding*. Springer, Berlin, Heidelberg, 2007, pp. 342–358.

[21] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2015, pp. 1–6.

[22] H. Farid, "Exposing digital forgeries from jpeg ghosts," *IEEE transactions on information forensics and security*, vol. 4, no. 1, pp. 154–160, 2009.

[23] P. Ferrara, M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "Unsupervised fusion for forgery localization exploiting background information," in *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, June 2015, pp. 1–6.

[24] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based cnn features," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1855–1864.

[25] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, March 2008.

[26] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, 2010.

[27] X. Kang, Y. Li, Z. Qu, and J. Huang, "Enhancing source camera identification performance with a camera reference phase sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 393–402, April 2012.

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

[29] 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, March 2007.

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

[31] 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.

[32] 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.

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

[34] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," in *Media Watermarking, Security, and Forensics*, 2017.

[35] T. Gloe and R. Böhme, "The 'dresden image database' for benchmarking digital image forensics," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 1584–1590. [Online]. Available: http://doi.acm.org/10.1145/1774088.1774427

[36] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, Oct 2017. [Online]. Available: https://doi.org/10.1186/s13635-017-0067-2

[37] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification," in *2012 IEEE International Conference on Acoustics, Speech and signal processing (ICASSP)*. IEEE, 2012, pp. 2257–2260.

[38] M. Brand, "Understanding manipulation in video," in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, Quarterly 1996, pp. 94–99.

[39] M. C. Stamm, W. S. Lin, and K. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1315–1329, 2012.

[40] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6226–6230.

[41] W. Chuang, H. Su, and M. Wu, "Exploring compression effects for improved source camera identification using strongly compressed video," in *2011 18th IEEE International Conference on Image Processing*, Sep. 2011, pp. 1953–1956.

[42] K. Kurosawa, K. Kuroki, and N. Saitoh, "Ccd fingerprint method-identification of a video camera from videotaped images," in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 3, Oct 1999, pp. 537–540 vol.3.

[43] E. K. Kouokam and A. E. Dirik, "Prnu-based source device attribution for youtube videos," *Digital Investigation*, vol. 29, pp. 91 – 100, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287618304377

[44] B. Hosler, O. Mayer, B. Bayar, X. Zhao, C. Chen, J. A. Shackleford, and M. C. Stamm, "A video camera model identification system using

TABLE X
CONFUSION MATRIX OF CAMERA MODEL IDENTIFICATION SYSTEM'S SINGLE-PATCH ACCURACY WHEN TRAINED ONLY ON H.264 ENCODED VIDEOS.

	M00	M01	M02	M03	M04	M05	M06	M08	M09	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35
M00	93.9	-	-	0.2	-	0.2	-	-	0.3	-	0.4	0.9	0.7	-	0.1	-	-	-	-	0.8	0.2	-	1.4	-	-	0.2	0.1	0.3	-	-	0.2	-	0.1	-
M01	0.3	76.9	4.6	-	3.4	2.1	-	-	0.1	0.2	-	0.2	1.2	-	-	7.5	0.1	0.6	-	-	-	-	1.4	0.3	0.1	-	-	-	-	0.8	0.1	-	0.1	-
M02	0.6	0.2	84.2	0.6	5.3	4.8	-	-	0.9	-	-	0.3	0.5	-	-	-	-	-	-	-	1.1	-	1.2	-	-	-	-	-	-	0.2	-	-	0.1
M03	-	-	0.5	96.7	0.5	-	1.3	-	-	-	-	-	0.4	-	-	-	-	-	-	-	0.2	-	0.1	-	-	-	-	-	0.1	-	0.2	-	-	-
M04	-	0.4	6.1	0.1	77.5	13.8	-	-	0.8	-	-	0.1	-	-	-	0.1	-	0.2	-	-	0.1	-	0.7	-	-	-	-	-	-	-	-	-	-	0.1
M05	-	-	1.4	-	7.0	89.3	-	-	0.5	-	-	0.2	-	-	-	0.2	-	0.1	-	0.1	-	-	0.9	-	-	0.2	0.1	-	-	-	-	-	-	-
M06	-	-	-	0.4	0.4	-	98.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.3	-	-	-	-	-	-	-	-	-	-	-
M08	-	-	-	-	-	-	-	92.2	-	1.3	-	-	-	-	-	-	-	-	0.6	-	2.8	-	-	-	-	-	0.1	0.5	-	-	0.7	-	1.8	-
M09	-	-	0.1	-	0.1	0.1	0.5	-	98.4	-	-	-	0.1	0.1	-	-	-	-	-	-	-	-	0.1	-	-	0.2	-	0.1	-	-	-	-	-	0.2
M10	2.1	-	-	0.3	0.1	0.1	-	1.4	-	69.1	0.9	0.5	0.8	-	1.8	0.1	0.2	0.2	9.1	1.4	6.9	-	0.5	-	-	-	-	1.2	-	-	2.5	-	0.8	-
M11	0.2	-	-	-	0.1	-	-	-	-	1.4	85.3	-	-	-	0.6	-	0.1	0.1	6.8	0.1	0.5	0.2	0.2	-	0.1	-	0.3	1.0	1.9	-	1.1	-	-	-
M12	1.9	-	0.7	0.2	0.5	-	0.1	-	1.4	0.1	-	69.8	16.3	-	0.7	0.4	0.1	0.1	0.1	0.5	0.2	0.1	1.5	0.1	-	4.4	0.3	-	-	0.2	0.2	-	-	0.1
M13	8.32	-	0.16	0.64	0.32	0.16	0.48	-	-	0.32	-	3.2	81.92	-	0.16	-	-	-	-	-	-	-	2.24	-	-	1.92	-	-	-	-	-	-	-	0.16
M14	-	-	-	-	-	-	-	-	-	-	-	-	-	95.3	-	-	-	-	-	-	-	-	-	1.9	1.0	-	-	-	-	-	-	1.8	-	-
M15	1.4	-	0.7	0.2	-	-	-	0.8	0.6	1.2	-	6.7	1.0	-	60.8	1.5	9.1	0.3	0.1	2.4	-	-	0.4	-	-	0.5	0.2	2.2	-	0.4	6.8	-	2.7	-
M16	0.3	12.8	3.9	-	0.3	0.7	-	-	0.1	0.1	0.3	0.9	0.2	-	0.4	62.4	0.3	5.7	0.1	0.2	-	0.5	6.2	-	-	-	0.2	-	-	2.4	1.8	-	-	0.2
M17	0.8	-	-	-	-	-	-	0.1	-	0.2	0.1	2.0	0.1	-	8.6	-	66.3	0.2	-	5.0	-	0.1	0.1	-	-	-	0.6	8.0	0.2	0.3	3.1	-	4.2	-
M18	0.1	0.9	0.1	-	-	0.2	-	-	0.9	-	0.6	1.6	-	-	0.2	9.0	1.5	74.6	-	0.7	-	0.4	2.6	-	-	0.2	0.7	0.8	0.7	1.1	1.4	-	-	1.7
M19	0.5	-	-	0.2	0.1	-	0.5	1.0	-	16.8	16.7	-	0.1	-	0.2	-	-	-	48.8	0.1	4.0	-	0.1	-	-	-	-	3.2	2.3	-	5.4	-	-	-
M20	1.2	-	0.6	0.1	0.4	0.2	-	-	-	0.6	-	0.3	3.1	-	0.2	1.2	-	-	0.2	81.1	1.0	-	1.5	-	-	2.5	4.0	-	0.1	-	0.1	-	1.6	-
M22	-	-	0.8	0.5	0.2	-	0.6	0.1	-	0.3	-	-	0.1	-	-	-	-	-	0.2	-	96.5	-	-	-	-	-	0.4	-	-	-	-	-	0.3	-
M23	-	-	-	-	0.9	-	-	-	2.4	-	2.1	0.4	0.6	0.3	0.1	-	0.1	0.6	0.8	0.1	-	68.9	0.6	-	0.4	4.9	4.3	1.2	4.0	5.4	0.3	0.3	-	1.3
M24	3.3	0.5	0.2	0.1	0.1	1.7	-	-	0.2	-	-	0.1	4.8	-	-	0.6	-	0.1	-	0.6	-	0.2	86.5	-	-	0.5	-	-	-	-	0.1	-	-	0.4
M25	-	-	-	-	0.7	-	-	-	-	-	-	-	-	1.5	-	-	-	-	-	-	-	-	-	96.7	0.5	0.1	-	-	-	-	-	0.4	-	0.1
M26	-	-	-	-	-	-	-	-	-	-	-	-	-	1.7	-	-	-	-	-	-	-	-	-	0.2	96.2	-	-	-	-	-	-	1.9	-	-
M27	2.3	-	-	0.9	0.5	-	0.3	-	1.9	-	0.1	2.1	7.8	-	0.1	0.1	0.8	0.4	-	0.7	0.4	2.8	2.7	-	0.2	71.3	3.6	-	-	0.4	0.3	-	0.1	0.2
M28	4.8	-	-	-	0.1	-	-	0.1	0.2	0.3	0.5	1.1	2.2	-	1.0	0.4	0.5	0.1	-	1.3	0.3	0.5	2.2	0.1	-	5.1	75.6	1.4	0.1	0.1	0.3	-	1.3	0.4
M29	0.4	0.2	-	0.3	-	-	0.1	1.9	-	3.9	6.6	1.0	0.1	0.1	2.5	-	2.7	1.9	0.7	2.1	1.1	0.4	0.1	-	0.1	0.4	1.9	53.5	5.4	0.8	9.9	-	1.2	0.7
M30	-	-	-	-	-	-	0.4	1.1	0.2	1.0	3.2	0.3	-	0.1	0.1	-	-	-	0.4	0.6	0.9	0.7	-	-	-	0.4	0.1	23.8	64.2	0.3	1.5	-	0.6	0.1
M31	-	1.2	1.2	-	0.1	0.4	-	-	1.2	-	-	3.5	0.2	-	0.2	4.4	-	7.6	-	0.2	-	2.2	3.1	-	-	0.9	0.3	-	0.8	67.3	1.4	-	-	3.8
M32	1.0	0.1	-	-	0.3	-	0.2	4.0	-	3.4	0.1	-	-	-	1.1	-	0.4	-	2.3	0.4	1.9	-	0.6	-	0.1	-	1.2	5.7	1.5	0.8	73.7	-	1.2	-
M33	-	-	-	-	0.1	-	-	-	-	-	-	-	-	4.3	-	-	-	-	-	-	-	-	-	1.3	5.2	0.1	-	-	-	-	-	89.0	-	-
M34	0.8	0.1	0.4	0.5	-	-	0.3	1.3	-	1.5	-	2.1	2.0	-	9.0	-	2.3	-	-	2.5	0.1	-	0.2	0.1	-	0.3	0.7	1.9	-	-	0.5	-	73.4	-
M35	-	-	-	-	0.5	-	0.2	-	2.3	-	-	0.2	-	-	-	-	-	0.8	-	0.2	-	0.1	1.6	0.1	-	0.2	0.7	0.1	-	1.7	0.5	-	-	90.8

deep learning and fusion," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8271–8275.

[45] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, Vigo, Galicia, Spain, 2016, pp. 5–10.

[46] X. Ding, Y. Chen, Z. Tang, and Y. Huang, "Camera identification based on domain knowledge-driven deep multi-task learning," *IEEE Access*, vol. 7, pp. 25 878–25 890, 2019.

[47] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid lstm and encoder-decoder architecture for detection of image forgeries," *IEEE Transactions on Image Processing*, pp. 1–1, 2019.

[48] 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.

[49] 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.

[50] Q. Phan, G. Boato, R. Caldelli, and I. Amerini, "Tracking multiple image sharing on social networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8266–8270.

[51] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus, "Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, Jan 2019, pp. 63–72.

[52] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

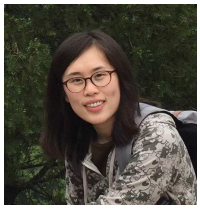
[53] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv e-prints*, p. arXiv:1412.6572, Dec 2014.

[54] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://arxiv.org/abs/1312.6199

[55] A. Costanzo and M. Barni, "Detection of double avc/hevc encoding," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 2245–2249.

Brian Hosler (S'19) is a Ph.D. student in the Department of Electrical and Computer Engineering at Drexel University, Philadelphia, PA, USA. He received a B.S. degree in electrical engineering in 2018 from Drexel University.

From 2017 to 2018 he worked as an Engineer at BMW manufacturing in Greenville, SC, USA. From 2014 to 2016 worked in the field of 2D nanomaterials at the Drexel Nanomaterials Institute. Currently, he is a research assistant in the Multimedia and Information Security Lab (MISL) at Drexel University, where he is conducting research on video and multimedia forensics. His current research interests include signal processing and machine learning.



Xinwei Zhao (S'15) is a Ph.D. candidate in the Department of Electrical and Computer Engineering at Drexel University, Philadelphia, PA, USA. She received a B.S. degree in electrical engineering in 2012 from Shandong University of Science and Technology, Qingdao, Shandong, China, and a M.S in electrical engineering in 2015 from Drexel University, Philadelphia, PA, US. Currently, she is a research assistant in the Multimedia and Information Security Lab (MISL) at Drexel University, where she is conducting research on image and multimedia forensics. His research interests include signal processing, multimedia forensics and anti-forensics, and machine learning.

Owen Mayer (S'12) is a Ph.D. candidate in the Department of Electrical and Computer Engineering at Drexel University, Philadelphia, PA, USA. He received a B.S. degree in electrical engineering in 2013 from Case Western Reserve University, Cleveland, OH, USA. From 2013 to 2014 he worked as a Staff Scientist at Ocean Acoustical Services and Instrumentation Systems, Inc. in Lexington, MA, USA. Currently, he is a research assistant in the Multimedia and Information Security Lab (MISL) at Drexel University, where he is conducting research on image and multimedia forensics. His research interests include signal processing, multimedia, and machine learning.

Chen Chen (S'16) received the B.Sc. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, China, in 2014. She is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at Drexel University, USA. She was a summer intern at Microsoft Research in 2013 and Facebook in 2018. Her study and research are in signal processing, multimedia forensics and information security.

James A. Shackleford (S'03-M'13) was born in Memphis, TN, USA, in 1983. He received the B.S. degree in electrical engineering from Drexel University, Philadelphia, PA, USA in 2006, the M.S. degree in electrical engineering from Drexel University in 2009, and the Ph.D. degree in computer engineering from Drexel University in 2011.

He was a Postdoctoral Fellow in the Radiation Oncology Department at Massachusetts General Hospital, Boston, MA, in 2012 and presently holds an appointment as an Adjunct Assistant Professor of Medicine in the Perelman School of Medicine at the University of Pennsylvania, Philadelphia, PA, USA, from 2017 to 2020. He is currently an Associate Professor of computer engineering at Drexel University. His research has been concerned with computer vision algorithm development as it applies to medical image processing for image-guided radiotherapy, with focus on segmentation and deformable registration techniques.

Dr. Shackleford is a member of the Association for Computing Machinery. He received the prestigious NSF CAREER Award in 2016 for the work entitled, "Low Latency, Parallel, and Context Aware Vision in Computed Tomography." He has authored a chapter in Nvidia's GPU Computing Gems book series on the topic of deformable registration and is the lead author of the book, "High Performance Deformable Image Registration Algorithms for Manycore Processors," published by Morgan Kaufmann in 2013.

Matthew C. Stamm (S'08–M'12) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 2004, 2011, and 2012, respectively.

Since 2013 he has been an Assistant Professor with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, USA. He leads the Multimedia and Information Security Lab (MISL) where he and his team conduct research on signal processing, machine learning, and information security with a focus on multimedia forensics and anti-forensics.

Dr. Stamm is the recipient of a 2016 NSF CAREER Award and the 2017 Drexel University College of Engineering's Outstanding Early-Career Research Achievement Award. He was the General Chair of the 2017 ACM Workshop on Information Hiding and Multimedia Security and is the lead organizer of the 2018 IEEE Signal Processing Cup competition. He currently serves as a member of the IEEE SPS Technical Committee on Information Forensics and Security and as a member of the editorial board of IEEE SigPort. For his doctoral dissertation research, Dr. Stamm was named the winner of the Dean's Doctoral Research Award from the A. James Clark School of Engineering. While at the University of Maryland, he was also the recipient of the Ann G. Wylie Dissertation Fellowship and a Future Faculty Fellowship. Prior to beginning his graduate studies, he worked as an engineer at the Johns Hopkins University Applied Physics Lab.