International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 7 Issue 3, May-June 2023 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Broadcasting Forensics Using Machine Learning Approaches

Amit Kapoor[1], Prof. Vinod Mahor[2]

[1]M Tech Scholar, [2]Assistant Professor,
[1,2]Computer Science & Engineering, Millennium Institute of Technology & Science, Bhopal, Madhya Pradesh, India

ABSTRACT

Broadcasting forensic is the practice of using scientific methods and techniques to analyse and authenticate Multimedia content. Over the past decade, consumer-grade imaging sensors have become increasingly prevalent, generating vast quantities of images and videos that are used for various public and private communication purposes. Such applications include publicity, advocacy, disinformation, and deception, among others. This paper aims to develop tools that can extract knowledge from these visuals and comprehend their provenance. However, many images and videos undergo modification and manipulation before public release, which can misrepresent the facts and deceive viewers. To address this issue, we propose a set of forensics and counter-forensic techniques that can help establish the authenticity and integrity of Multimedia content. Additionally, we suggest ways to modify the content intentionally to mislead potential adversaries. Our proposed tools are evaluated using publicly available datasets and independently organized challenges. Our results show that the forensics and counter-forensic techniques can accurately identify manipulated content and can help restore the original image or video. Furthermore, in this paper demonstrate that the modified content can successfully deceive potential adversaries while remaining undetected by state-of-the-art forensic methods.

KEYWORDS: *Broadcasting Forensics, Multimedia Content Analysis, Image and Video Manipulation Authenticity and Integrity, Forensics and Counter-Forensic Techniques, ML*

How to cite this paper: Amit Kapoor | Prof. Vinod Mahor "Broadcasting Forensics Using Machine Learning Approaches" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-7 | Issue-3, June 2023, pp.1034-1045, URL: www.ijtsrd.com/papers/ijtsrd57545.pdf

IJTSRD57545

1. INTRODUCTION

The widespread use of smartphones has fundamentally altered how we communicate with one another. Unprecedented levels of digital content creation, such as photographs and films, are now possible because to these gadgets. The gold standard for storing our memories or indicating our participation at social occasions is currently thought to be digital information. Numerous pieces of software have been created in order to edit and improve these digital assets because of the value placed on images and videos as well as their significant role in social networks and journalistic Broadcasting. This software innovation has a cost, though. Image and video manipulation, whether done for aesthetic or malevolent reasons, is becoming a popular practise. The public is starting to reject taking any image or video as digital evidence because of the abundance of these faked Broadcasting. One of the first to make a suggestion of this pattern. The "fake news" problem [1] and the recent rise in popularity of machine learning-based techniques like generative adversarial networks that may be abused by bad actors are also aggravating this scepticism.

The research community has been creating methods to check and authenticate such content in order to reduce the spread of false information and assist law enforcement authorities with the completion of investigations in which digital assets may be involved. These initiatives gave rise to the area of digital image and video forensics [2]. The present state of digital image and video forensics includes the two significant study topics of source image forensics and content-based forensics, which are the subject of this work. We specifically frame the issue of source image forensics as a source camera identification challenge, which we suggest resolving using CNN-based techniques. Computer graphics forensics, on

which we specifically focus on the current Deepfakes video manipulation approach, are a subset of content-based forensics. We also research the production and detection of hostile content images for CNN camera model identification techniques. The use of video metadata analysis for tampering detection is something else we look into [3].

The analysis of the picture's statistics and other pattern recognition techniques were initially used to solve the issue of source image forensics. In the field of source image forensics, deep learning-based techniques have been effectively used, and they have demonstrated their efficacy in a number of contests [4]. In order to better comprehend the outcomes produced by deep learning techniques, we provide a thorough analysis of those methods in the lines that follow, as well as their interrelationships, as seen in Figure 1.1 Any reader of this work will be able to better understand the context in which we developed our solutions as a result of this.

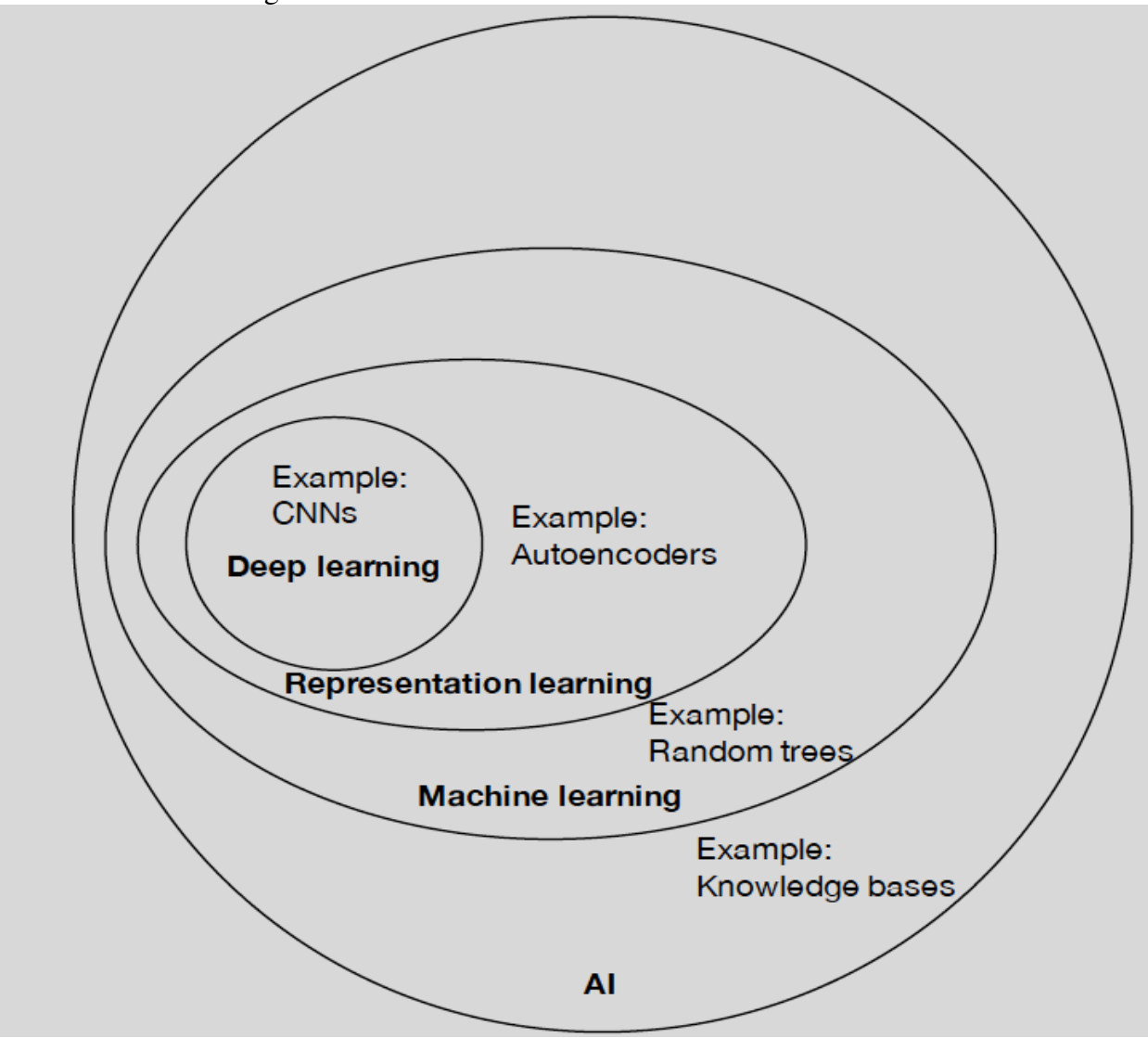


Figure 1.1: The Venn diagram demonstrating the relationship between deep learning and representational learning.

Deep neural networks became practical in thanks to a number of innovations, which also gave rise to the area of deep learning. Following deep learning-based techniques has consistently produced outstanding results on a variety of tasks, including handwritten text recognition, understanding video scenes, and image classification. Convolutional neural networks, recurrent neural networks, and generative adversarial networks are some of the specialised methods in computer vision that have been developed to address these problems. Among these, CNNs and RNNs have been demonstrated to be efficient when handling image- and video-related tasks, and as a result, they have been used as the foundation for several digital forensics techniques [5–6].

Convolutional layers, pooling layers, and activation functions are the primary fundamental elements of CNNs. These layers are often layered together to create the convolutional neural network's architecture. The four important recent developments in the construction and training of CNNs are regularisation, structural reformulation, and loss function. The most crucial of these for enhancing performance is structural reformulation [7].

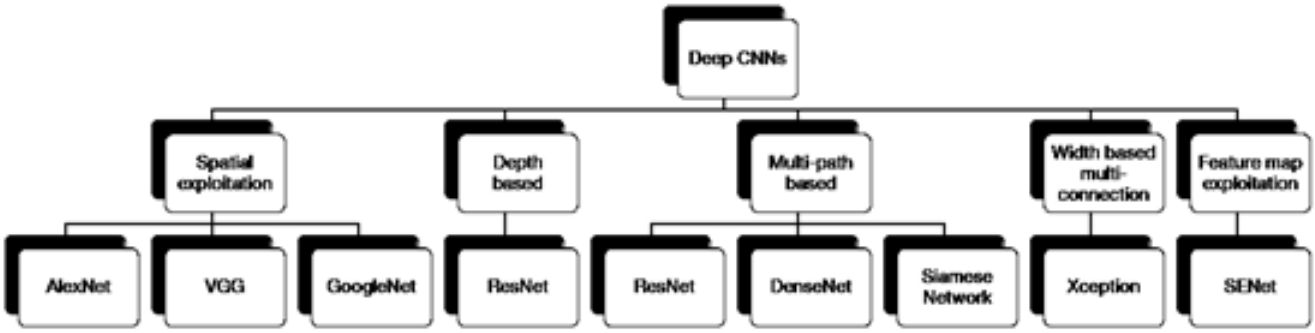


Figure 1.2: Deep CNN architecture taxonomy.

1.1. Description of the Camera Model

Identification of the camera model is essential for blindly determining the source and validity of a picture. Modern methods make use of characteristics from the image acquisition pipeline that describe the traces that various camera models (such as those left by proprietary demos icing techniques) left on pictures. A convolutional neural network (CNNs) might be used for camera model recognition in light of the very accurate performance attained by feature-based approaches as well as the advancements brought about by deep architectures in machine learning. We especially look at (i) how well different network topologies can learn discriminant characteristics from seen pictures, and (ii) how much training data is necessary to obtain a certain level of accuracy [8].

2. LITERATURE SURVEY

The study and identification of unlicensed broadcast content, known as broadcasting forensics, is a challenge that is getting harder to solve in the modern world. Due to its capacity to automatically analyse and discover abnormalities in broadcast information, the usage of machine learning algorithms has grown in popularity as a remedy for this issue. We will examine some of the most recent studies on the application of machine learning to broadcasting forensics in this review of the literature.

He and Zhang released one of the first publications in this area in 2015, outlining a machine learning-based method for locating unauthorised films on social Broadcasting sites. Based on a collection of data taken from the video frames, the authors utilised a supervised learning technique called Support Vector Machine (SVM) to categorise films as authorised or unauthorised. A lot of data is available on social Broadcasting, which is growing essential for communication, and this platform might be useful for analysis. We use Twitter to tweet about seemingly unrelated subjects and various emotions at specific times. In this article, sentimental analysis is suggested as a way to learn about people's thoughts or feelings. Every tweet is divided into its suitable emotion. Either a pleasant or negative feeling might be present. The two phases of the suggested technique are pre-processing and categorization. After completing all essential pre-processing, the corpus is formed. For classification, techniques including Logistic Regression, Linear SVC, Bernoulli NB, Decision Tree Classifier, Voting Classifier, and KNN Classifier are employed. Data from Twitter for 2020 and 2021 has been collected for testing. Linear SVC performs better in terms of accuracy on training data, whereas linear regression performs better in terms of accuracy on testing data [7-8].

Chiang and associates proposed a comparable strategy for identifying unapproved audio material in radio broadcasts in 2022. Based on their spectrogram representations, the authors employed a Convolutional Neural Network (CNN) to categorise audio portions as authorised or unauthorised. We need to gather artefacts from many systems, applications and databases, network and security devices, and other things as part of digital forensics and incident response. The suggested technique obtained an accuracy of over 95% in detecting unauthorised audio segments. Artefacts will assist us in comprehending contemporary and historical events that took place in a certain system at a specific period. Event Logs and Registry Values, particularly USB, which offers details on external Broadcasting linked to the computer, and Run Keys, which provides details on the virus persistence mechanism, are two of these artefacts on Windows machines. It's conceivable for an attacker to run a malicious power shell script that controls the system or to leave a back door power shell script that allows access to the victim's system. In order to investigate if attackers are employing power shell, we are providing a method to gather power shell events. The first step in identifying a danger or an attack in its entirety is to examine security event logs. As a result, if a user receives a spam email, security events will produce a few associated events, and we will also use the SIEM tool for additional analysis. Our technology also converts all event logs to text files, making it simple for an investigator to review those files in a SIEM tool. All monitoring

events, such as the beginning and ending of processes, network connections, any modifications to file creation, timestamps, etc., are recorded in the Sysmon event log. The sysmon event logs may be fetched by this Python-based utility and saved in a text file. A researcher can submit the file to the SIEM tool for additional study. This may provide the current circumstance a new security patch [9-10].

3. PROBLEM IDENTIFICATION

In this section we introduce the problem formulation with the notation used throughout the chapter. We then provide the reader a brief overview about CNNs and their use in Multimedia Forensics.

3.1. Problem Formulation

Let us consider a color image I acquired with camera model l belonging to a set of known camera models L. In this chapter, we consider the patch-based closed-set camera model attribution problem as presented in [6]. Given an image I, this means:

Select a subset of K color patches Pk, k ∈ [1, K].

Obtain an estimate ˆlk = C(Pk) of the camera model associated with each patch through a camera attribution function C.

Optionally obtain final camera model estimate ˆl through majority voting over ˆlk, k ∈ [1, K].

Our goal is to detect whether a patch Pk is a good candidate for camera model attribu- tion estimation. To this purpose, we propose a CNN architecture that learns a function G expressing the likelihood of a patch Pk to provide correct camera model identification, i.e., gk = G(Pk). High values of gk indicate high probability of patch Pk to provide correct camera information. Conversely, low gk values are attributed to patches Pk that cannot be correctly classified. Pixel-wise likelihood is then represented by means of a reliability map M, showing which portion of an image is a good candidate to estimate image camera model, as shown in Figure 3.1.

3.2. Convolutional Neural Networks in Multimedia Forensics

In this section, we present a brief overview of the foundations of convolutional neural networks (CNNs) that are needed to follow the chapter. For a thorough review on CNNs, we refer the readers of this paper.

Deep learning and in particular CNNs have shown very good performance in several computer vision applications such as visual object recognition, object detection and many other domains such as drug discovery and genomics [11]. Inspired by how the human vision works, the layers of a convolutional network have neurons arranged in three dimensions, so each layer has a width height, and depth. The neurons in a convolutional layer are only connected to a small, local region of the preceding layer, so we avoid wasting resources as it is common in fully-connected neurons. The nodes of the network are organized in multiple stacked layers, each performing a simple operation on the input. The set of operations in a CNN typically comprises convolution, intensity normalization, non-linear activation and thresholding, and local pooling. By minimizing a cost function at the output of the last layer, the weights of the network are tuned so that they are able to capture patterns in the input data and extract distinctive features. CNNs enable learning data-driven, highly representative, layered hierarchical image features from sufficient training data.

Figure 3.1: A block schematic illustrating the suggested strategy. Image I'm divided into patches.

There has been a growing interest in using convolutional neural networks in the fields of image forensics and steganalysis [12]. These papers mainly focus on architectural design of CNNs where a single CNN model is trained and then tested in experiments. Data- driven models have recently proved valuable for other Multimedia forensic applications as well, Moreover, initial exploratory solutions targeting camera model identification show that it is possible to use CNNs to learn discriminant features directly from the observed known images, rather than having to use hand-crafted features. As a matter of fact, the use of CNNs also makes it possible to capture characteristic traces left by non- linear and hard to model operations present in the image acquisition pipeline of capturing devices.

Results have in- dicated that learning from interBroadcastingte representation in CNNs instead of output probabil- ities, and then jointly retraining the final architecture, leads to performance improvement [13].

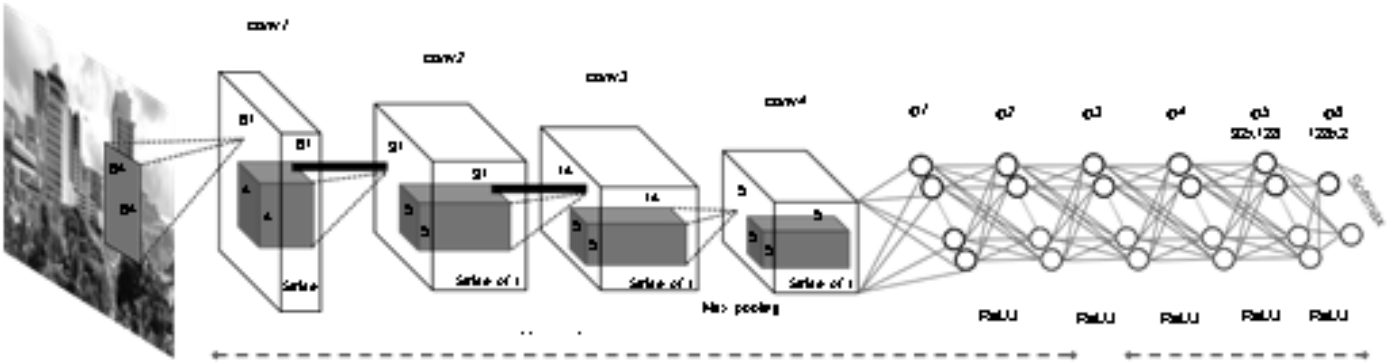


Figure 3.2: Representation of the proposed CNN architecture M engaged in image patch processing

3.3. Estimating Patch Reliability Method

In this part, we go into further depth about how we estimate patch reliability and attribute cameras. The following phases make up the proposed pipeline (see Figure 3.2):

A. Patches are separated from the image being analysed.
B. To calculate the likelihood of patch reliability, a CNN is utilised.
C. We estimate a camera model for each patch using the same CNN.
D. The camera is attributed to the entire image, and a dependability mask is created.
E. A thorough description of each step is provided below.
F. Patch Extraction

The suggested technique examines picture patches to function. The colour picture I is initially divided into a collection of K patches Pk, k [0, K]. The resolution of each patch is 64 by 64 pixels. Depending on the dimension, the patch extraction stride might range from 1 to 64.Mr. Sri Kalyan Yarlagadda, Prof. Fengqing Maggie Zhu, Prof. Paolo Bestagini, and Prof. Stefano Tubaro collaborated on this project.

4. PROPOSED METHOD

The proposed counter-forensic method. Our method consists of an adversarial image generator module that can be added to a CNN- based camera model evaluation pipeline. In Figure 4.2, we assume a similar structure to the previously presented pipeline in Section 4.2. Our adversarial image generator module takes as input the set of K patches that have been extracted from the image I that is being analyzed. When presented with new image patches, our module can work in two different modes.

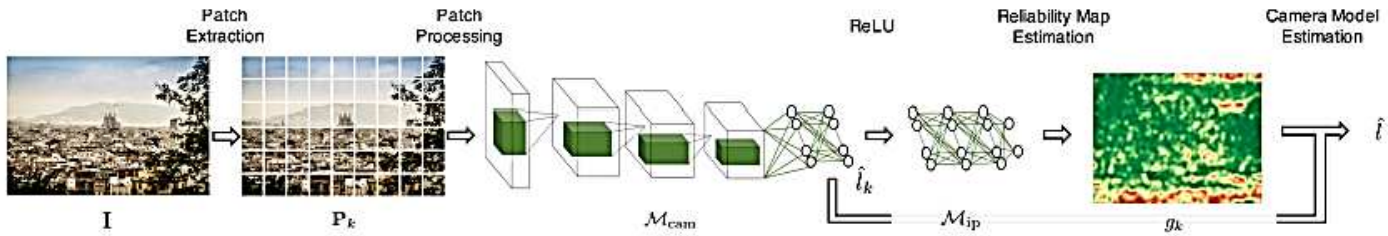


Figure 4.1: Block diagram of our proposed method.

In the first operation mode, the adversarial image generator module does an untargeted image manipulation, that is, it does not try to perturb the image patches to produce a specific misclassification class. Instead, we use the derivative of the loss function of the CNN with respect to the input image patches to add a perturbation to the images. The derivative is computed using backpropagation with the labels L^kJ, k ∈ [1, K] that are given by the CNN detector when it first processes the unmodified image patches. This procedure is known as the fast gradient sign method (FGSM) [13-14].

In the second operation mode, the adversarial image generator module does a targeted image manipulation. In this case, we try to perturb the image patches to produce a specific misclassification class LJ, different from the true real label L that is associated with the analyzed image I and its associated Pk patches. In this mode of operation, we exploit the 1This is joint work with Dr. Yu Wang, Dr. Luca Bondi, Prof. Paolo Bestagini, and Prof. Stefano Tubaro forward derivative of a CNN to find an adversarial perturbation that will force the network to misclassify the image patch into the target class by computing the adversarial saliency map. Starting with an unmodified image patch, we perturb each feature by a constant offset ϵ. This process is repeated iteratively until the target misclassification is achieved. This procedure is known as the Jacobian-based saliency map attack (JSMA) [15].

Its present a detailed overview of both FGSM and JSMA techniques as follows.

4.1. Fast Gradient Sign Method

In, the fast gradient sign method was introduced for generating adversarial exam- ples using the derivative of the loss function of the CNN with respect to the input feature vector. Given an input feature vector (e.g. an image), FGSM perturbs each feature in the direction of the gradient by magnitude ϵ, where ϵ is a parameter that determines the pertur- bation size. For a network with loss J(Θ, x, y), where Θ represents the CNN predictions for an input x and y is the correct label of x, the adversarial example is generated as

$$x^* = x + \epsilon \text{sign}(\nabla_x J(\Theta, x, y))$$

With small ϵ, it is possible to generate adversarial images that are consistently misclassified by CNNs trained using the MNIST and CIFAR-10 image classification datasets with a high success rate [16-18].

4.2. Jacobian-Based Saliency Map Attack

In [58], an iterative method for targeted misclassification was proposed. By exploit- ing the forward derivative of a CNN, it is possible to find an adversarial perturbation that will force the network to misclassify into a specific target class. For an input x and a convolutional neural network C, the output for class j is denoted Cj(x). To achieve an output of target class t, Ct(x) must be increased while the probabilities Cj(x) of all other classes j /= t decrease, until t = arg maxj Cj(x). This is accomplished by exploiting the adversarial saliency map, which is defined as

$$S(x,t)[i] = \begin{cases} 0, \text{if } \frac{\partial C_t(x)}{\partial x_i} < 0 \text{ or } \sum_{j \neq t} \frac{\partial C_j(x)}{\partial x_i} > 0 \\ (\frac{\partial C_t(x)}{\partial x_i}) | \sum_{j \neq t} \frac{\partial C_j(x)}{\partial x_i} |, \text{otherwise} \end{cases}$$

for an input feature i. Because we work with images in this chapter, in our case each input feature i corresponds to a pixel i in the image input x. Starting with a normal sample x, we locate the pair of pixels {i, j} that maximize S(x, t)[i] + S(x, t)[j], and perturb each pixel by a constant offset ϵ. This process is repeated iteratively until the target misclassification is achieved. This method can effectively produce MNIST dataset examples that are correctly classified by human subjects but misclassified into a specific target class by a CNN with a high confidence [19-20].

5. RESULT DISCUSSION

The dataset splitting approach suggested in to our issue, we assess our solution in this study. This approach has been specifically designed for the Dresden Image Dataset, which comprises of 73 devices from 25 distinct camera model families. Each camera shoots a different quantity of pictures. Every angle offers a different reason for shooting.

A scene is a grouping of a certain motivation and a specific location. Keeping this criterion in mind, there are 83 total scenes in the dataset. We only take into account camera models with more than one instance since we are attempting to categorise picture patches at the level of camera model rather than instance level. This results in a total of 18 camera types (since the only real difference between the Nikon D70 and D70s is their on-device screen) and around 15,000 photographs.

The dataset, which consists of 18 camera models, is divided into three sets: training DT, validation DV, and evaluation. DE. DT is once more divided into two equal groups, Dcam T and D ip T. While D ip T is used to train Mip, Dcam T is used to train the parameters of Mcam.

To prevent overfitting, DV is utilised to determine how many epochs to employ during training. The trained network is then fairly evaluated using DE on a disjoint collection of pictures.

Avoiding overfitting the data is crucial while training a CNN. Our objective is to identify the camera model from an image in our dataset, which contains pictures of various scenarios captured by various cameras. It is crucial that DT and DV differ enough from one another since DV is utilised to prevent overfitting. Additionally, it is crucial that we test using data that differs from the training set. We take the following actions in order to fulfil these objectives:

- A single instance per camera and a collection of 11 scenarios are used to choose the images for DE.
- Images for DT are chosen from 63 distinct situations and extra camera instances.
- For DV, images are chosen from the remaining 10 scenes using the same camera instances as for DT.

This separate sets DV and DT in terms of scenes, resulting in strong training.

K = 300 non-overlapping colour patches with a size of 64 by 64 are taken from each image for training, validation, and testing. There are more than 500 000 patches in the generated dataset Dcam T, which are divided into 18 classes. According to the Mcam classification findings, D ip T is lowered to 90 000 patches to balance dependable and unreliable image patches. Finally, there are more than 700 000 and 800 000 patches in DV and DE, respectively.

5.1. Instructional Techniques

We suggest a two-tiered transfer learning-based strategy called Transfer since the suggested approach relies on a pretrained network (i.e., Mcam). We also examine the Scratch and Pre-Trained extra tactics for comparison's purposes. Following, we provide information on each technique in more depth.

Scratch. The most straightforward training method is this one.

The whole two-class architecture M is trained using simply D ip T for training and DV for validation. This might be regarded as a fundamental training approach. We employ the Adam optimizer with batch size set to 128 and default parameters as indicated in. Binary-cross entropy loss is the default.

Pre-Trained. This tactic benefits from the potential use of a M_{cam} that has already been trained. In this instance, we use the output of SoftMax normalisation to train M_{cam} for camera model attribution. Validation is done on DV, while training is done on D_{cam} T. After M_{cam} has been trained, we freeze its weights and use D_{ip} T and DV to train the remaining components of the architecture M_{ip} as a two-class classifier (i.e., reliable vs. non-reliable patches). The Adam optimizer is used during both training phases to optimise, employing batches of 128 patches and default values. Our loss function is categorical-cross entropy.

Transfer. This two-tiered training approach aims to fully utilise the suggested architecture's capacity for transfer learning. The first phase is training M_{cam} for camera model attribution using D_{cam} T and DV as datasets, SoftMax normalisation on its output, and. With default settings, 128 patches per batch, and categorical-cross entropy as the loss function, Adam is used to optimise this training phase.

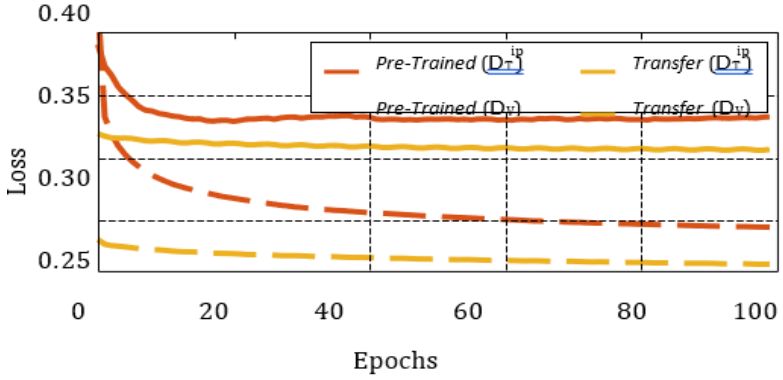
The second stage of M training entails freezing all of $M_{cam's}$ convolutional layers and continuing to train all of M_{cam} and $M_{ip's}$ inner product layers using the datasets D_{ip} T and DV. This makes it possible to concurrently learn the classifier $M_{ip's}$ weights and adapt the feature extraction process in $M_{cam's}$ last two layers (i.e., ip1 and ip2) to the classification problem. For this stage, we utilise stochastic gradient descent (SGD) with a fluctuating learning rate between 5 and 15 10 5 as the optimizer and binary-cross-entropy as the loss metric.

This decision is inspired by exploratory research done in [17] and empirically supported in our study.

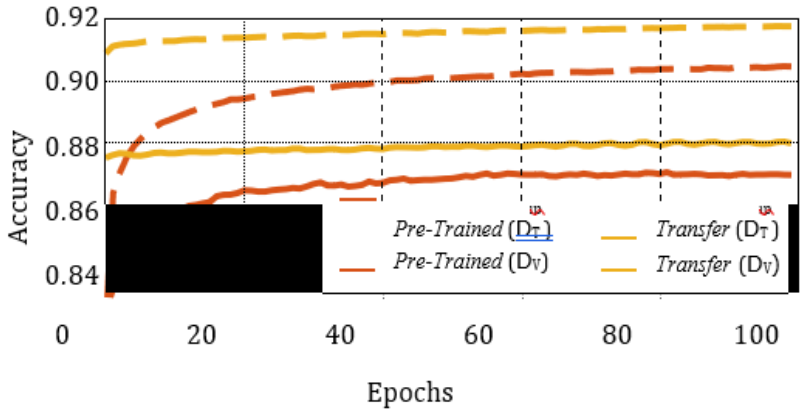
Its explain the experimental findings in this section. The first demonstrate how the suggested method can tell which patches include camera model information and which ones are unsuitable for this job. Then, the demonstrate how the suggested technique may be used to enhance camera model recognition.

5.2. Patch Reliability

Architecture on CNN. The initial round of tests focused on selecting the network architecture for Mip. To do this, we used the pre-Trained method to train a variety of architectures over the course of 15 epochs. We chose all conceivable arrangements of up to six inner product layers (with ReLU activation), each made up of 32, 64, or 128 neurons, as our layouts. Two neurons are always set as the last layer, followed by softmax. The model Mip with the best validation accuracy for each tested number of layers was chosen from this experiment, and it is M^2 ip comprised of two inner product layers with 128 and 2 neurons, respectively.



(a) Loss

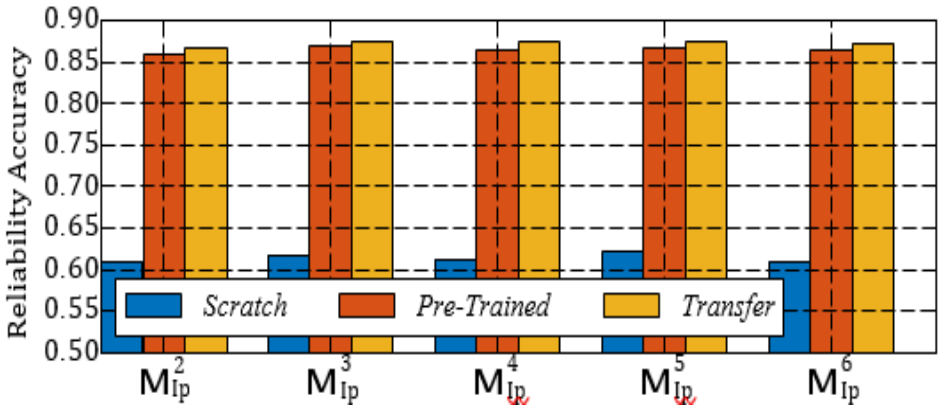


(b) Accuracy

Figure 5.1: Loss and accuracy curves for the M4 Pre-Trained and Transfer techniques on training (Dip) and validation (DV) datasets.

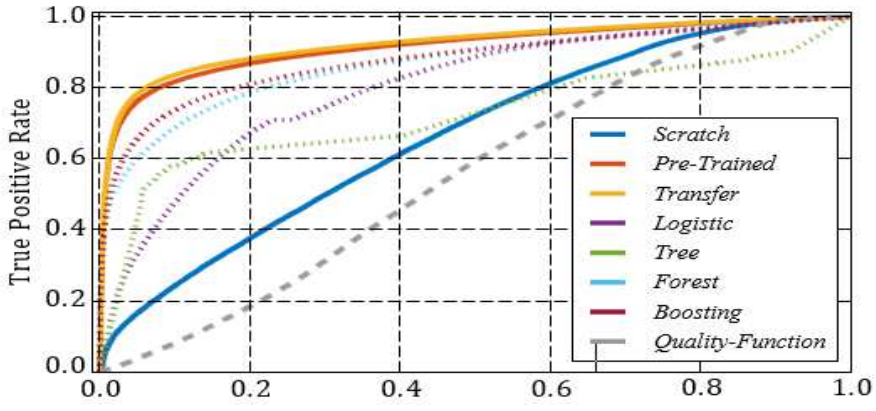
- M_{Ip}^{3} composed by three inner product layers with 64, 128 and 2 neurons, respectively.
- M_{Ip}^{4} composed by four inner product layers with 64, 32, 128 and 2 neurons, respectively.
- M_{Ip}^{5} composed by five inner product layers with 64, 32, 64, 128 and 2 neurons, respectively.
- M_{Ip}^{6} composed by six inner product layers with 64, 32, 32, 64, 64 and 2 neurons, respectively.

Loss and accuracy curves on training (Dip T) and validation (DV) datasets using Pre-Trained and Transfer strategies on M_{Ip}^{6}.



CNN Architecture

Figure 5.2: The Transfer approach yields the most precise results for any design.



False Positive Rate

Figure 5.3: ROC curves for accurate patch recognition. Solid lines are used to depict the suggested techniques. Additional baselines are dashed or dotted

Baseline evaluation. We also took into consideration two potential baseline solutions in order to better validate the suggested strategy.

Table 5.1: Using just the credible patches from the test dataset, one may accurately attribute camera models

M_{ip}	Strategy	Patches	Accuracy	Acc. Delta
M^2_{ip}	*Scratch*	553 475	0.9009	0.0342
	Pre-Trained	618 958	0.9478	0.0811
	Transfer	637 135	**0.9511**	**0.0844**
M^3_{ip}	*Scratch*	518 228	0.9041	0.0374
	Pre-Trained	626 767	0.9520	0.0853
	Transfer	641 808	**0.9554**	**0.0885**
M^4_{ip}	*Scratch*	562 897	0.8963	0.0296
	Pre-Trained	649 515	0.9499	0.0832
	Transfer	647 998	**0.9530**	**0.0864**
M^5_{ip}	*Scratch*	511 425	0.9045	0.0378
	Pre-Trained	648 665	0.9529	0.0862
	Transfer	651 508	**0.9520**	**0.0853**
M^6_{ip}	*Scratch*	517 386	0.9035	0.0367
	Pre-Trained	651 405	0.9501	0.0834
	Transfer	652 308	**0.9541**	**0.0867**

The first involves utilising several supervised classifiers and using the 18-element vector that Mcam returned as a feature. For this, we trained a gradient boosting classifier (Boosting), a decision tree (Tree), a random forest (Forest), and a logistic regressor (Logistic). After performing parameter grid-search training on D, we applied z-score feature normalisation to each approach and chose the model that achieved the best validation accuracy on DV. On evaluation set DE, accuracy findings for patch reliability were 70.7%, 73.9%, 78.6%, and 81.8%, respectively. None of them come close to making up the 86% of the suggested answer. The quality-function described in [18] (Quality- Function) is the second baseline solution that we examined. This function calculates the suitability of each patch for training Mcam, and it outputs a value between 0 and 1 for each patch. We decided that a comparison was required for thoroughness even though Quality-Function was not meant to serve as a test reliability indicator. Figure 3.7 depicts receiver operating characteristic (ROC) curves that were generated by thresholding our reliability likelihood estimation gk, the soft output of the other classifiers (such as the logistic regressor, decision tree, etc.), and the quality-function for this purpose.

There are always more chosen patches in E when employing the Transfer approach (bold) than there are in E overall. Accuracy has risen by more than 8% compared to random patch selection. yielded the value [7]. As anticipated, using the quality-function described in [7] yields less precise results. In contrast, when trained using the Transfer technique, the suggested method outperforms every other classifier.

5.3. Attribution of Camera Model

We first examined the suggested method's ability to choose trustworthy patches, and then we looked at how it affected camera model attribution. For this reason, we present in Table 3.1 the evaluation set findings for the three training procedures and the five Mip models that were evaluated when a single patch was used to attribute camera models.

- Patches, or the expected total of valid patches.
- Accuracy, or the typical result of camera model attribution.
- Accuracy Delta, or the improvement in accuracy compared to choosing patches at random rather than employing patch selection.

These findings show that a greater than 8% improvement in camera model attribution is feasible.

Figure 5.4 displays the confusion matrix findings for assessment data DE randomly picking patches using Mcam. 87% accuracy on average per patch. Figure 3.9 displays the same outcomes when using M4 to evaluate just patches that the re deemed credible. The accuracy rises to above 95% in this case. Comparing the two figures reveals that only the application of trustworthy patches can correct numerous erroneous classifications outside of the confusion matrix diagonal.

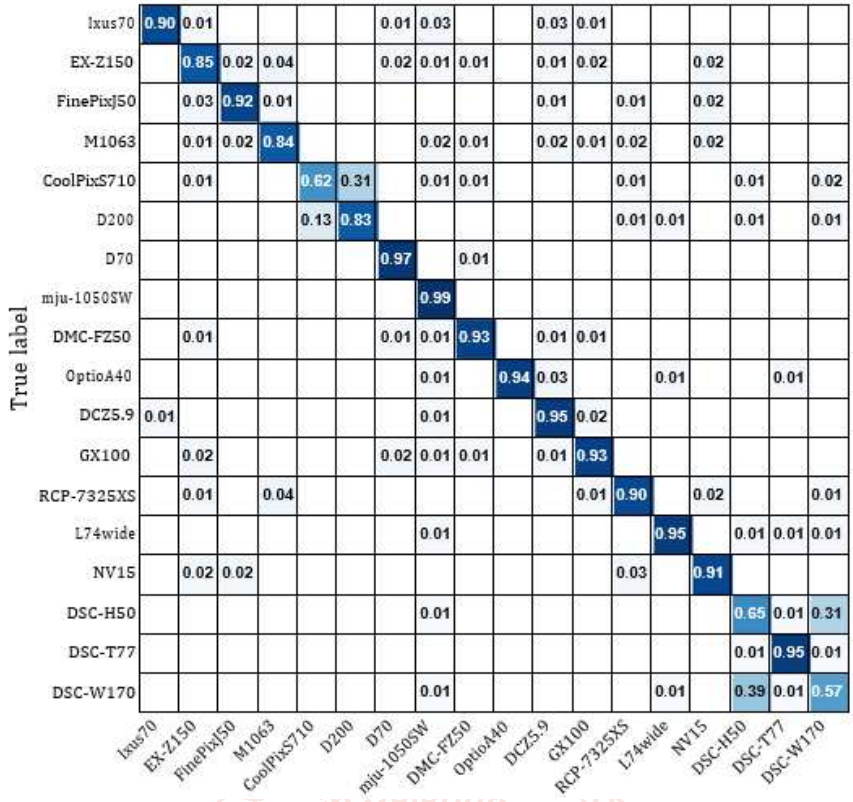


Figure 5.4: Camera model attribution confusion matrix obtained with Mcam on DE without patch selection

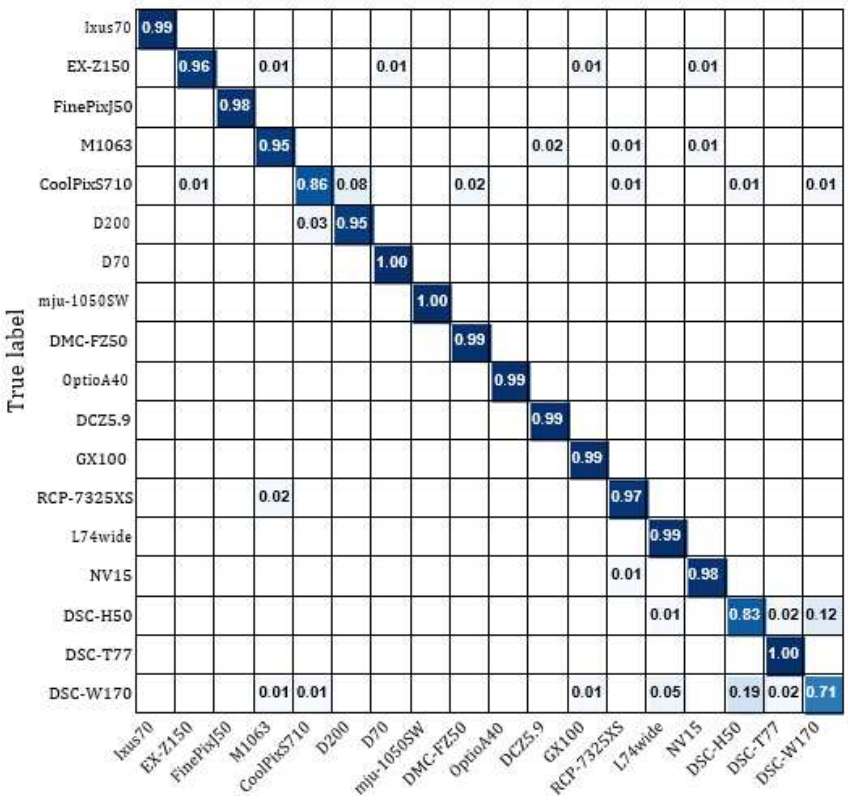


Figure 5.5: Confusion matrix for the camera model's attribution produced using Mcam on DE using patches chosen.

The suggested technique provides a reliability mask that emphasizes which picture areas are regarded as dependable in terms of camera attribution in addition to the effect on camera model attribution. This might be helpful in the future to identify which visual features camera model attribution CNNs value more highly. Additionally, it may be used in conjunction with localization techniques based on camera model traces to splice out potentially inaccurate regions from the study.

6. CONCLUSION

The effectiveness of identifying video modifications can be significantly increased with appropriate data and the use of simple machine learning classifiers. Previous efforts in detecting video manipulation have primarily focused on examining pixel data to identify fabricated content. Our approach involves the combination of a random forest and an SVM, trained on characteristics of Multimedia streams derived from both genuine and falsified videos. This method has proved highly successful in detecting video alterations, even with limited data. Our ongoing research aims to develop techniques that automatically sanitize data based on these findings. Specifically, the plan to remove auxiliary header information and metadata that could potentially reveal sensitive information about the video's source.

In future. possible future work could involve extending the machine learning approaches used in broadcasting forensics to more complex video manipulation scenarios. For instance, exploring how deep learning models such as convolutional neural networks (CNNs) can be used to identify more sophisticated tampering techniques, such as deepfake videos or subtle manipulations that are hard to detect using traditional approaches.

Another area of research could be the development of real-time video manipulation detection systems that can be used by broadcasters to ensure the integrity of live content. This would require exploring how to implement machine learning models with low latency and high accuracy on live video streams.

REFERENCES

[1] L. N. Smith, "Cyclical learning rates for training neural networks," Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 464–472, Mar. 2017, Santa Rosa, CA. [Online]. Available: https://doi.org/10.1109/WACV.2017.58

[2] K. Liang, J. K. Liu, R. Lu, and D. S. Wong, "Privacy concerns for photo sharing in online social networks," IEEE Internet Computing, vol. 19, no. 2, pp. 58–63, Mar. 2015. [Online]. Available: https://doi.org/10.1109/MIC.2014.107.

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

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, Jun. 2016, Las Vegas, NV. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[5] Huang, Z. Liu, K. Q. The inberger, and L. van der Maaten, "Densely connected convolutional networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Aug. 2016, Honolulu, HI. [Online]. Available: https://doi.org/10.1109/CVPR.2017.243

[6] J. Thies, M. Zollho¨fer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2387–2395, Jun. 2016, Las Vegas, NV. [Online]. Available: https://doi.org/10.1109/CVPR.2016.262

[7] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, pp. 5–10, Jun. 2016,

[8] Vigo, Spain. [Online]. Available: https://doi.org/10.1145/2909827.2930786

[9] Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," IEEE Signal Processing Letters, vol. 23, no. 5, pp. 708–712, May 2016. [Online]. Available: https://doi.org/10.1109/LSP.2016.2548421

[10] L. Baroffio, L. Bondi, P. Bestagini, and S. Tubaro, "Camera identification with deep convolutional networks," arXiv:1603.01068, Mar. 2016. [Online]. Available: http://arxiv.org/abs/1603.01068

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826, Jun. 2016, Las Vegas, NV. [Online]. Available: https://doi.org/10.1109/CVPR.2016.308

[12] G. Huang, Z. Liu, K. Q. The inberger, and L. van der Maaten, "Densely connected convolutional networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Aug. 2016, Honolulu, HI.

[Online]. Available: https://doi.org/10.1109/CVPR.2017.243

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1800–1807, Jul. 2017, Honolulu, HI. [Online]. Available: https://doi.org/10.1109/CVPR.2017.195

[14] G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," Pattern Recognition, vol. 77, no. C, pp. 354–377, May 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.10.013

[15] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," ACM Transactions on Graphics, vol. 36, no. 4, pp. 95:1–95:13, Jul. 2017. [Online]. Available: https: //doi.org/10.1145/3072959.3073640

[16] P. Mullan, D. Cozzolino, L. Verdoliva, and C. Riess, "Residual-based forensic comparison of video sequences," Proceedings of the IEEE International Conference on Image Processing, pp. 1507–1511, Sep. 2017, Beijing, China. [Online]. Available: https://doi.org/10.1109/ICIP.2017.8296533

[17] X. Ding, G. Yang, R. Li, L. Zhang, Y. Li, and X. Sun, "Identification of motion-compensated frame rate up-conversion based on residual signals," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 7, pp. 1497– 1512, Jul. 2018. [Online]. Available: https://doi.org/10.1109/TCSVT.2017.2676162

[18] R. Raghavendra, K. B. Raja, S. Venkatesh, and C. Busch, "Transferable deep-cnn features for detecting digital and print-scanned morphed face images," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1822–1830, Jul. 2017, Honolulu, HI. [Online]. Available: https://doi.org/10.1109/CVPRW.2017.228

[19] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1831–1839, Jul. 2017, Honolulu, HI. [Online]. Available: https://doi.org/10.1109/CVPRW.2017.229

[20] L. Bondi, S. Lameri, D. Gu¨era, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based cnn features," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1855–1864, Jul. 2017, Honolulu, HI. [Online]. Available: https://doi.org/10.1109/CVPRW.2017.232