Video tampering localisation using features learned from authentic content.

JOHNSTON, P., ELYAN, E., JAYNE, C.

2020

Neural Computing and Applications (2020) 32:12243–12257
https://doi.org/10.1007/s00521-019-04272-z

ENGINEERING APPLICATIONS OF NEURAL NETWORKS 2018

Video tampering localisation using features learned from authentic content

Pamela Johnston[1] · Eyad Elyan[1] · Chrisina Jayne[2]

Received: 11 January 2019 / Accepted: 21 May 2019 / Published online: 30 May 2019

Abstract
Video tampering detection remains an open problem in the field of digital media forensics. As video manipulation techniques advance, it becomes easier for tamperers to create convincing forgeries that can fool human eyes. Deep learning methods have already shown great promise in discovering effective features from data, particularly in the image domain; however, they are exceptionally data hungry. Labelled datasets of varied, state-of-the-art, tampered video which are large enough to facilitate machine learning do not exist and, moreover, may never exist while the field of digital video manipulation is advancing at such an unprecedented pace. Therefore, it is vital to develop techniques which can be trained on authentic or synthesised video but used to localise the patterns of manipulation within tampered videos. In this paper, we developed a framework for tampering detection which derives features from authentic content and utilises them to localise key frames and tampered regions in three publicly available tampered video datasets. We used convolutional neural networks to estimate quantisation parameter, deblock setting and intra/inter mode of pixel patches from an H.264/AVC sequence. Extensive evaluation suggests that these features can be used to aid localisation of tampered regions within video.

Keywords CNN · Compression · Video tampering detection · Deep learning

1 Introduction

Automated video analysis is an increasingly important area of research. Video content creates a unique visual record, but not all aspects of video content are apparent to human eyes and this is of particular relevance in today's age of fake news and falsified video. Machine learning techniques are already used to alter video content by changing weather conditions via style transfer [1] or by performing digital re-enactment [2, 3]. In discriminating between authentic content and digital re-enactment using recent techniques, human assessors did little better than random guessing [4]. There is an increasing urgency to develop techniques to detect evidence of video processing even when it is invisible to human eyes. This raises the important question: How do we develop useful features for visual data when we might not be able to perceive such features using our own biological sensors? Deep learning provides a good tool kit for feature discovery from data; however, it is necessarily data hungry. In fields such as video tampering, a large, labelled and sufficiently varied dataset which encompasses multiple examples from many recent techniques does not yet exist, although [4] and its recent successor [5] show great promise. In fast moving fields, a complete dataset may never exist as, in the time taken to gather and label the data, many more new and improved techniques will be discovered. Therefore, we must develop new techniques to exploit features common to many data examples.

Video compression is prevalent in digital society. The vast majority of online video has been compressed using lossy formats such as H.264/AVC [6] or MPEG2 [7]. Compression formats have been designed with the human visual system in mind, and the effects remain largely below the threshold of detection for human eyes. It has been shown that compression does impact classification performance of convolutional neural network (CNN) classifiers

✉ Pamela Johnston
p.a.johnston3@rgu.ac.uk

[1] Robert Gordon University, Aberdeen, UK

[2] Oxford-Brookes University, Oxford, UK

[8, 9] and pre-existing compression in original source images may have caused these effects to be understated. CNN classifiers are passively affected by compression; therefore, it is reasonable to use them to actively detect the level of compression directly from pixels. Moreover, accurate estimation of compression parameters, such as quantisation, could be used to enhance the performance of CNN classifiers across differing quality levels.

Video tampering techniques are growing at an unprecedented rate [10]. Detection methods can be active or passive [11, 12], but, since many existing videos are unwatermarked at source, passive detection methods are more applicable. Passive tampering detection can be categorised into recompression, region tampering and inter-frame forgery [11]. Region tampering includes copy–move attacks where copied regions can come from the same frame in the video, similar to image copy–move [13] or from a different frame in the same video [14]. Variations on region tampering include: splicing where content from two different sources is spliced together and inpainting where an object or region is removed from the sequence and the removal concealed. Inter-frame forgery is where an integer number of frames is added, deleted or reordered. Regardless of the editing method, however, any tampering at the pixel level of a compressed video requires recompression of the video bitstream [15, 16], and detection of compression parameters from the pixels themselves will evidence recompression. Compression parameters can provide underlying evidence of how a video has been processed. For example, two videos might exhibit different compression parameter distributions which remain in evidence when they are spliced together.

An intuitive indication of recompression is where the Quantisation Parameter (QP) encoded within the bitstream fails to match the value estimated from the pixels. This is most obvious to human eyes when the bitrate and syntax elements of the bitstream imply high-quality video data, but the pixel content exhibits visible compression artifacts such as blockiness. Accurate QP estimation from pixels may also aid tampering detection in other ways such as key frame identification and QP distribution analysis. In order for this to happen, objective methods of measuring QP directly from video sequence pixels are required. An ideal QP estimator would also operate accurately over small patches to enable localisation of tampered regions which is an advancing area of research [12, 17]. For singly compressed frames, estimated QP can be verified by encoded bitstream syntax elements. In multiply compressed video, there will be mismatches between estimated QP and syntax elements, and differing QP patterns may be detected over spatially or temporally tampered regions.

This work extends the work in [18] and takes a step towards utilising compression parameters derived directly from the pixels themselves. The main contributions are:

- We show CNNs can be trained to estimate different compression parameters such as quantisation parameter, intra- or inter-frame type and deblocking filter setting for standalone sequence patches with reasonable accuracy.
- We combine our CNN models along with frame deltas to identify key frames in encoded sequences. Performance is evaluated on singly and doubly compressed sequences with varying bitrates.
- We use our CNN models on existing tampered video datasets [4, 19, 20] to demonstrate that some tampered video sequences, particularly spliced content, exhibit distinct compression profiles and that these can be used to localise tampered regions.

2 Related work

There are a number of challenges in the detection of tampered video. Although tampering detectors exist, these are often tailored to specific tampering techniques. The authors of [4] produced one of the largest manipulated datasets to date, based exclusively on the digital re-enactment strategy of [3]. A deep neural network was successfully trained to detect manipulated video content with less than 1% error where humans did little better than guessing, but it was shown in [21] that this learning does not necessarily transfer readily to other video manipulation methods. In their recent paper reviewing video content authentication techniques, Singh and Aggarwal [15] noted that there is no consistent database of realistically doctored videos. A large dataset specifically for image rebroadcast detection was produced in [22]. They demonstrated how previous techniques, which had achieved good results on small, specific datasets, were significantly outperformed on this large, diverse dataset by a CNN, which obtained over 97% accuracy in determining which images had been rebroadcast and which were authentic. Manipulation techniques are currently more powerful than detection techniques [23], with many ways to digitally alter an image or video but relatively few methods to detect such manipulations. There is therefore a need to develop detection techniques that are independent of the type of video manipulation.

Machine learning techniques are evidently very good at discovering consistencies and patterns within data and used to detect tampering [4, 22], but novel techniques are required to fulfil their large data requirements. In [23], a Siamese neural network was used to identify whether pixel

patches exhibited consistent image metadata and could, therefore, have come from the same image pipeline and be part of the same authentic image. The network was trained using only authentic images and their associated EXIF metadata, so a large dataset could be gathered quickly and simply. Using 80 features from EXIF metadata and 3 further processing techniques (JPEG compression level, Gaussian blur and re-scaling), the authors managed to classify whether two 128 × 128 pixel patches were consistent with each other and thus achieve a new state of the art in image tampering localisation.

While image metadata are often available in online image files, authentic video metadata are not as readily available. Video files are much larger and will often be edited or compressed and recompressed for storage or streaming purposes. Recompression can also mask tampering, but some evidence can remain. Any processing can leave a forensic fingerprint on the pixels of a video sequence. Analysing this fingerprint can provide evidence of video tampering, such as splicing, inpainting or inter-frame tampering. Here we look specifically at aspects of video compression that can usefully contribute towards this fingerprint. In uncompressed video, sensor pattern noise can be utilised to identify a camera model as in [24], and identification of two different camera models in the same frame can be used to infer tampered video. Shullani et al. [25] compiled a dataset of authentic video and found that sensor pattern noise was affected by the compression applied by two different social media platforms. Moreover, the two different social media platforms (YouTube and WhatsApp) had different effects on the data, implying that their compression mechanisms are distinct from each other. In [26], elements of compression, such as macroblock compression type, were used to detect inter-frame tampering. Using machine learning techniques, deleted frames in an MPEG-2 encoded video sequence were detected with 95% accuracy. However, given that the compression features were extracted directly from the bitstream itself, [26] could be simply defeated via recompression. This paper aims to overcome that challenge by estimating compression parameters directly from the pixels. These patterns are then used to identify areas of inconsistency which could infer video tampering.

The human visual system is adequate to detect some compression effects and can quantify "no reference" quality [27, 28]. The source of video compression visual effects can be found by examining transformations used in compression standards. A video sequence comprises key (intra-) frames, which provide access points into the sequence, and predicted (inter-) frames which rely on data from previously encoded frames. Key frames contain more data than predicted frames and are sometimes compressed more to meet bit rate requirements, occasionally resulting in visible artifacts. As noted in [11], many inter-frame tampering detection methods assume perfectly periodic key frames and struggle to detect tampering that aligns with key frames. Identification of key frames directly from pixels is a strong feature in inter-frame tampering detection, but intra/inter decisions are not only made at the frame level.

In H.264/AVC and MPEG-2, frames are further divided into macroblocks which are blocks of 16 × 16 pixels. Each macroblock can be intra- or inter-coded. Intra-frames can only contain intra-macroblocks, but inter-frames can contain both intra- and inter-macroblocks. For non-predicted data, the pixel data itself are transformed into the frequency domain using Discrete Cosine Transforms (DCTs), quantised and variable length encoded for transmission. For predicted data, a suitable patch of reference pixels is located; then, the *difference* between current and reference data is transformed, quantised and encoded. Quantisation is performed as in Eq. 1 where δ is DCT coefficients of a macroblock or residual, C is the compressed coefficients and Q_s represents the quantisation step as indexed by the quantisation parameter [29].

$$C = round\left(\frac{\delta}{Q_s}\right) \tag{1}$$

Higher QP indexes larger Q_s and means more frequency coefficients are filtered out entirely. An increase in QP often manifests visually as an increased "blockiness", that is, discrete regions of macroblocks consisting single or few frequency coefficients. Most often, low frequencies have higher signal amplitudes, so sharp edges persist while textures are reduced. In key frames, macroblock edges align uniformly within the frame. This visual effect was more apparent in earlier video compression standards [7] where non-integer DCTs forced regular inclusion of key frames. Periodic key frames limited rounding error drift between encoder and decoder but were sometimes visible as a pulse in the sequence as accumulated rounding errors were reset. The integer transforms introduced in H.264/AVC [6] reduced the role of key frames to access points in the bitstream and consequently reduced the periodic pulse in video sequences. HEVC [30] defines other techniques to reduce visible compression artifacts but is yet to be fully adopted. H.264/AVC is more common in the wild. Compression artifacts are not restricted to artificial block edges, however, and can also manifest as a lack of specific frequency detail or as banding in areas of smooth colour/intensity transition.

As noted in [11], many inter-frame tampering detection methods struggle to detect tampering that aligns with key frames. Methods fail when a complete Group of Pictures (GOP) from one key frame to the next is deleted, added or temporally moved. It can be deduced from this that these

techniques ultimately rely on a detected mismatch between key frames identified using features from the pixels and either those derived directly from bitstream syntax elements, or those estimated from assumed encoder behaviour. It is clear from this that there are some differences between intra- and predicted frames in video compression.

As part of an investigation into using deep neural networks to determine image quality, Bosse et al. [27] developed a method to estimate QP of HEVC frames directly from pixels. They achieved accurate results for average QP estimation over a complete frame using a patch-wise technique and dataset synthesised from UCID [31]. The method was applied to key frames only. QP estimation was framed as a regression problem, and the dataset used to train the network contained labelled patches compressed with all possible QPs. Although averaged QP prediction for a complete frame was accurate, a heatmap showing individual patch contributions displayed great variation between patches.

This work examines QP estimation in the context of patches taken from H.264/AVC video sequences. We also look at identification of key frames from the pixels themselves. H.264/AVC is currently one of the most popular video compression standards and is used on YouTube, broadcast video and public datasets. A CNN is trained to classify frame patches from a video sequence using their quantisation parameters as labels. Unlike [27], we also investigate whether these features can be used to detect tampering in videos.

3 Proposed framework

The full framework is summarised in Fig. 1. In order to implement the framework, the following techniques are required:

- CNNs trained to estimate QP, inter/intra frame mode and sequence deblocking filters from the pixels
- A method to calculate frame deltas
- A method to identify key frames
- A method to localise tampering

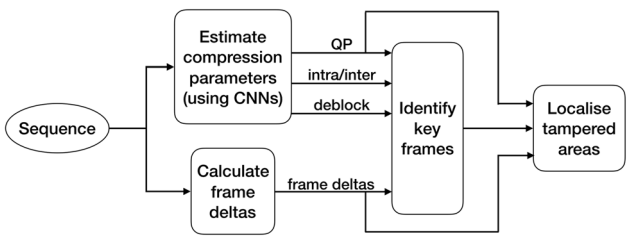


Fig. 1 Summary of the proposed framework

These techniques are detailed in the following subsections. First, we use authentic data to train CNN feature detectors. Then we use these feature detectors to express pixel patches from a video sequence as feature vectors. We use key frames only to increase the efficacy of the CNN compression feature detectors. The feature vectors are then clustered in to two clusters using k-means clustering [32]. This assumes that there are two different distributions present in the data, representing authentic and tampered data, and our experimental evaluation shows that this is a valid assumption for some tampered data.

3.1 Authentic datasets for CNN training

When examining the effects of compression, it is vital to start with unprocessed data. Standard YUV 4:2:0 sequences from xiph.org are commonly used for video compression quality analysis.[1] Strictly speaking, YUV 4:2:0 is a compressed format due to reduced colour channel resolution; however, it is widely used as a starting format in video compression. The sequences from xiph.org come in various dimensions and cover a wide variety of subjects from studio-shot sequences to outdoor scenes. All sequences are single camera, continuous scenes with varying degrees of camera motion.

A large amount of data are required to train a neural network, and uncorrelated data will produce a more generalised network. It is possible to use still image data as single frame sequences when focussing on spatial compression artifacts and excluding temporal compression. For this purpose, the images of UCID [31] were used. UCID consists of uncompressed images which are either 512 $\times$ 384 pixels or 384 $\times$ 512 pixels and cover a wide variety of subject matter. All are natural scenes and taken with the same camera. Of the original reported 1338 images in the dataset, only 882 were available.[2] Using a dataset of single images is not ideal since predicted frames cannot be examined. However, it allows for a greater variety of pixel combinations in a smaller dataset because individual images are uncorrelated. Each image from UCID was regarded as a single frame video sequence.

Following [18], we process the video using various compression parameters to synthesise a number of original datasets summarised in Table 1. Each video sequence was compressed using the open-source H.264/AVC encoder x264 and one of a range of constant QP levels using variable bitrate mode. Constant quantisation parameters were selected with an even distribution: QP = [0, 7, 14, 21,

[1] Available from Derf's Media Collection: https://media.xiph.org/video/derf.

[2] UCID images from http://jasoncantarella.com/downloads/ucid.v2.tar.gz.

Table 1 Uncompressed, authentic datasets for synthesising compression features

Name	Source	Length	Dimensions	Key frame	Deblock
AllVid	xiph.org	45 videos	176 × 144 to 1920 × 1080	1/250	Off
AllIntra	xiph.org	45 videos	176 × 144 to 1920 × 1080	All	Off
AllDeblock	xiph.org	45 videos	176 × 144 to 1920 × 1080	1/250	On
UCID	UCID [31]	882 images	512 × 384 or 384 × 512	All	O

Table 2 Patch datasets used for learning compression features

Name	Source	Label	# Train	# Test
IntraForQP	AllIntra	QP	764,640	56,392
Intra0vsInter1	AllIntra, AllVid	$I = 0, P = 1$	836,512	12,992
Deblock1	AllDeblock, AllVid	Deblock = 0.1	836,512	12,992

28, 35, 42, 49]. Constant bitrate rate control and psychovisual options were turned off. Deblocking filter was set as specified in Table 1. For datasets containing predicted frames, the key frame interval was 250. Patches were then extracted from the decoded YUV 4:2:0 sequences and upsampled from YUV 4:2:0 to YUV 4:4:4.

Table 2 summarises synthesised datasets. A large temporal stride was used to limit correlation between patches from the same video sequence. Consecutive frames are similar to each other, and a neural network trained with correlated dataset will be subject to overfitting. All datasets were prepared in advance of network training, and the original video sequences were split into disjoint train and test sets prior to compression and patch sampling to prevent data leakage.[3] The images of UCID were encoded as intra-frames and used as supplemental data to AllIntra.

A patch size of 80 × 80 pixels was used. Block edge artifacts in intra-frames will present themselves at macroblock (and sub-block) boundaries. Therefore, any patch size larger than 16 × 16 will capture block edges. In [27], a small patch size of 32 × 32 was selected, but results in [18] showed that a larger patch size yielded more accurate local results. When aligned with the macroblock grid, 80 × 80 pixels covers 5 × 5 complete macroblocks. A larger patch size allows for more context and image features within the patch to contribute towards feature classification. Spatial strides were selected so that there was no patch overlap in the training set, although patches taken from the same video sequence would exhibit some correlation.

Each dataset in Table 2 consists of a number of YUV 4:4:4 patches, each labelled appropriately. QP was labelled according to the quantisation parameter. The inter- or intra-labels depended only on the frame type, and not on individual macroblocks. The deblocking filter setting was done on a sequence level. From each of these synthesised datasets, a neural network was trained to perform classification according to the label.

[3] Training sequences: akiyo, bridge-close, bridge-far, carphone, claire, coastguard, foreman, hall, highway, mobile, mother-daughter, paris, silent, stefan, tennis, waterfall, old_town_cross, crowd_run, ducks_take_off, in_to_tree, mobcal, old_town_cross, parkrun, shields. Test sequences: bus, flower, news, tempete.

3.2 Network architecture

In [18], three different network architectures (NAs) were examined for one compression parameter (QP). Here, one fully convolutional architecture was used, similar to the architecture used in [33] which obtained particularly good results on CASIA2 [34]. CASIA2 is known to suffer from asymmetric image processing between tampered and non-tampered image classes [35]; therefore, this network architecture is already known to perform well in detecting image processing. Each network was trained on only one compression parameter, yielding three sets of network weights, one each for QP, deblock and inter/intra.

The architecture used was: conv5×5-30, norm, pool2×2, conv3×3-16, conv3×3-16, conv3×3-16, norm, pool2×2, conv3×3-16, conv3×3-16, softmax. A stride of 2 for convolutions allowed sufficient overlap to encounter compression artifacts while reducing the number of network parameters. Image patches of format YUV 4:4:4 were scaled to values between 0 and 1 and whitened. In order to preserve compression artifacts in situ, no further data augmentation were used. Batch size was 128 patches. Adam was used for gradient descent [36] in the quantisation parameter network and stochastic gradient descent for the intra/inter and deblock features. The networks were implemented using TensorFlow.

3.3 CNN compression parameter estimation accuracy

The quantisation parameter (QP) in H.264/AVC can be expressed as:

$$0 \leq QP \leq 52, \; QP \in \mathbb{R} \tag{2}$$

QP relates directly to Q_s in Eq. 1. Patches with similar QP labels exhibit similar compression features, and confusion matrices produced by the network reflected this. Different QPs might have very similar effects on a given patch, depending on patch content. A patch of solid colour, for example, transforms to a single high amplitude, low frequency coefficient which is nonzero on quantisation. Such an extreme example is unlikely in natural scenes, but it demonstrates how applying close QPs might result in identical patches with different labels. Therefore, QP was restricted to [0, 7, 14, 21, 28, 35, 42, 49] in the synthesised datasets. Using all possible QP would generate an extremely large dataset with more potential for ambiguous examples and increase model training times. Another source of ambiguity is the presence of skipped macroblocks. In simple terms, a skipped macroblock in a predicted frame is identical to the reference macroblock in the reference frame. This means that there may exist some predicted regions whose pixel content is identical to regions in a key frame. Using larger patch sizes decreases this risk.

3.4 Key frame detection

One thing evident in [18] was that accurate estimation of quantisation parameters in predicted frames is challenging. This is because quantisation is applied to the *difference* between the motion compensated macroblock from previous encoded frames and the current macroblock. If this residual is unknown, as in the case where only the pixels can be relied on, then it is difficult to estimate the quantisation parameter. In order to avoid such challenges, it was decided to identify and process only key frames.

A large percentage of compression in video comes from predicted frames. It is much more efficient to compress the differences between frames than it is to compress every single frame in isolation. With the advent of integer transforms in standard compression codecs, periodic key frames are no longer required to correct transform rounding error accumulation. Therefore, it is reasonable to assume that key frames are in the minority in a video sequence. Moreover, because non-predicted frames are inherently larger than predicted frames and rate control mechanisms attempt to avoid peaks in bitrates, key frames are often compressed using a higher QP than predicted frames. Key frames can also exhibit more block artifacts than predicted frames. This can be used to distinguish key frames from predicted frames.

To identify key frames, we first estimated the quantisation parameter *qp*, the inter/intra parameter *ip* and the deblocking parameter *db* for patches in every frame of a sequence. The patches were 80 × 80 pixels and separated by a stride of 16 pixels (overlapping). Patch values for qp, ip and deblock were then averaged over each frame in a sequence and the differences between the averages taken. The three different predictions were then combined as in Eq. 3

$$a_f = (\overline{qp}_f - \overline{qp}_{f-1}) \times (\overline{ip}_f - \overline{ip}_{f-1}) \times (\overline{db}_f - \overline{db}_{f-1}) \quad (3)$$

where $\overline{qp}_f$ represents the average CNN predicted quantisation parameter for frame f and $\overline{qp}_{f-1}$ is the same parameter for the previous frame. Key frames were then defined as any frame where the value of a_f was more than two standard deviations from the mean of a_f.

3.5 Datasets for tampering detection

Three publicly available datasets were used for evaluation: FaceForensics [4], D'Avino et al. [19] and Video Tampering Dataset (VTD) [20].

FaceForensics [4] is a large, tampered video dataset consisting over 1000 videos where content is restricted to talking heads, including news readers, with minimum dimensions of 480p and 300 frames. The authentic source videos were originally scraped from YouTube, and the tampered sequences use a variant of Face2Face [3]. Every tampered video has an authentic counterpart, and the video sequences are supplied as losslessly compressed files. For these experiments, only the first frame of each of the sequences in the test set was used. Once the dataset was divided into patches, it exhibited a large imbalance with only 3% positive samples. In order to create an additional, balanced dataset, crops of the tampered areas and corresponding authentic areas were created by using the difference between related authentic and tampered sequences. Areas outside of the crops are pixel-wise identical between tampered and authentic content.

The dataset provided by D'Avino et al. [19] consists of 10 spliced videos. The sequences are all 720p and 281–488 frames in length. Each sequence is a single camera, continuous scene, although the camera is not static in all sequences and some sequences are subject to significant camera motion. The dataset provides uncompressed .avi video files for original, forged and binary mask for each sequence; however, the source videos used to create splices have been compressed in the past and evidence of compression can be found in the pixels (see Sect. 4.3). Again, the dataset has a large imbalance, with 4.2% of all patch samples labelled tampered. Original background videos were filmed by the authors, but content for chroma-keyed regions was obtained from YouTube. The dataset has been benchmarked by the authors using an auto-encoder-based method. The auto-encoder is trained on a short sequence of authentic frames so that predictions made by the auto-

encoder showed the greatest deviation in tampered regions. Our method does not require any authentic frames.

VTD [20] comprises 26 forged sequences and their 26 authentic counterparts. There are also 7 authentic sequences in the dataset, but these were discarded. The tampered video files comprise 10 sequences of spatio-temporal copy–move, 6 inter-frame tampering (frame shuffling) and 10 spliced sequences. The sequences are between 420 and 480 frames in length and are all available in 720p, barring a single 420p sequence. Some sequences contain cut scenes, and there is evidence of non-motion compensated resampling within the dataset, which implies that source videos were not pristine. The dataset is distributed via a YouTube channel and, as such, is subject to recompression. The videos were downloaded from YouTube selecting the highest possible bitrate and frame dimensions, and the average bitrate was 1.7 Mbps, which equates to a compression rate of 0.06 bits per pixel (bpp). Recompression itself makes mask extraction noisy and tampering localisation particularly challenging. The lack of mask provision for this dataset also highlights the somewhat philosophical question of whether a pixel which remains unchanged between authentic and tampered sequences, yet forms part of a tampered object, is considered tampered or not. However, data from the compressed bitstream are also available, allowing accurate identification of key frames from the most recent (YouTube) compression. VTD is, as yet, unbenchmarked.

For VTD, masks were extracted using a thresholded difference between each frame of the forged and corresponding authentic sequences. Pixels with a difference higher than the threshold were labelled tampered, and those below labelled authentic. Thresholds in the range 0–64 were selected manually for each sequence. The mask pixels were then filtered temporally, using majority vote across 3 frames consecutive frames to remove erroneous compression noise. Finally, morphological operations were applied to each mask frame for further clean up. Using these masks, less than 2% of dataset patches were labelled tampered.

3.6 Localisation of tampering

Tampering was localised within detected key frames only and used only predicted QP and frame deltas. Frame deltas were calculated for 16×16 pixel patches to correspond with the QP prediction values. The frame delta value was set to 1 whenever the mean absolute difference of a given 16×16 pixel patch and the co-located patch in the previous frame was nonzero, and set to zero otherwise.

Unsupervised clustering was used to group feature vectors representing pixel patches into one of two groups. For VTD and D'Avino, these feature vectors consisted predicted QP and frame deltas. For the datasets based on FaceForensics, frame deltas were unavailable, given that only the first frame in the sequence was used. Therefore, all three compression features were used in the feature vector. The resulting clusters were nominatively labelled "authentic" or "tampered". Given that some tampered video content is simply two or more authentic videos spliced together, these labels could be effectively switched to more closely match the ground truth on sequences containing spliced data.

Table 3 CNN results

Trained to classify	Number of classes	Accuracy (%)
QP	8	71.18
Inter/Intra	2	69.23
Deblock	2	66.53

For assessment, Matthews correlation coefficient (MCC, Eq. 4) and F1 score (Eq. 5) were used. MCC provides a score between -1 and 1 where 0 represents uncorrelated data, 1 is completely correlated data and -1 is completely inversely correlated. This is particularly useful for when classes can be flipped as in the case for spliced video.

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4}$$

$$F1 = \frac{2TP}{2TP + FN + FP} \tag{5}$$

Both of these metrics focus mainly on true positives and are relatively unsuitable for detection of inter-frame tampering. To account for this, we have used only the intra-frame tampered sequences of VTD. They are also both subject to class imbalance.

4 Experimentation and discussion

Results indicate that CNNs can be trained to achieve a reasonable level of accuracy in determining three compression parameters directly from pixels and that this accuracy is sufficient to identify key frames and aid localisation of tampering in some sequences. This demonstrates how authentic video can be used to fulfil the large data requirement of deep learning techniques even when the application is the detection of forged video.

4.1 CNN compression parameter estimation

The three trained CNNs achieved the accuracies listed in Table 3. Training a network to detect compression

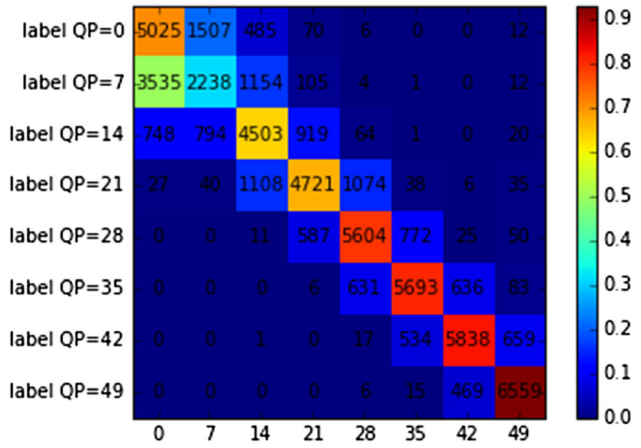


Fig. 2 Confusion matrix for QP-trained network

parameters directly from pixels will always be subject to some degree of error. Not all pixel regions in natural images will evidence all relevant frequencies necessary to unambiguously assign a given compression parameter. This effect can be seen in the confusion matrix (Fig. 2). The network incorrectly labels some lightly compressed patches as heavily compressed, and this is likely due to a natural lack of high frequencies in those regions. Similarly for the inter/intra network, we label on the frame level, not on the macroblock level. It is possible for some pixels to remain unchanged between key and predicted frames due to skipped macroblocks. With regard to the deblock filter, although it is set on or off on a sequence level, the parameters which control the level of filtering on individual macroblocks are controlled by motion vectors. All of these factors contribute to ambiguity in the labels for individual macroblocks, but are sufficiently evened out over a patch size of 80×80 pixels to achieve a workable level of accuracy. This is shown in Fig. 3 where the average estimated QP for the complete frame is accurate, but individual regions display some variance. In particular, the white sky region is allocated a relatively high QP, demonstrating a lack of high frequency coefficients.

4.2 Key frame identification

The key frame identification was performed as in Sect. 3.4. Because it is based on outliers, this method of key frame identification assumes at least one key frame in the sequence, other than the initial frame. Key frames occur when specified by the compression encoder. Cut scenes will sometimes trigger the encoding of a key frame, however not always, and not all key frames occur on cut scenes. Frame differences can sometimes indicate key frames, but they are not reliable as can be seen by comparing Figs. 4 and 5.

To gauge its efficacy, the method of key frame identification was tested on a number of sequences which comprised compressed and recompressed versions of YUV test sequences. These were first compressed with the open source encoder x264 using different bitrates (0.01, 0.02, 0.05, 0.1, 0.2 bpp) and an intra-frame frequency of 1/30. The resulting compressed sequences were then recompressed using the same bitrates but an intra-frame frequency of 1/25. It was found that the method performed very well at bitrates of 0.02–0.2 bpp in identifying the key frame from the latest compression. This is shown by the graph for single compression and the line for the second compression in the graph of double compression in Fig. 6. Below bitrates of 0.02 bpp, the visual video quality was very poor and the predicted quantisation parameter started to saturate to its highest level, leading to inaccuracies in key frame identification. Above bitrates of 0.2 bpp, the predicted quantisation parameter did not saturate, but accuracy still dropped. It is probable that the reduced accuracy was due to rate control choices made in the x264 encoder. With a higher bitrate available, peaks in bitrate due to key frames are comparatively reduced. Therefore, it becomes more efficient to encode key frames with higher quality, yielding more accurate reference frames and consequently reducing the bits required for predicted frames. Recompression at bitrates below 0.1 bpp effectively camouflaged key frames from the previous compression. As bitrates of the second compression process increased, evidence of key frames from the previous compression process emerged, as can be seen by the rise in the "first compression i-frames" F1 score graph.

The method of identifying key frames was then applied to the VTD dataset. It should be noticed that our method was effective at bitrates corresponding to those of VTD. As can be seen in the graphs in Fig. 7, combining predicted QP, inter/intra and deblocking values as in Eq. 3 provided a clear indication of key frames in the latest compression. Using the frame averaged mean absolute difference between frames yielded noisy results and did not accurately identify key frames. Comparing the key frames identified using this method with those extracted from the bitstreams of the forged sequences of VTD [20] achieved 87 true positives out of a total of 93 key frames. There were 27 false positives, giving an F1 score of 0.84. The majority of false positives (16 false positives) came from two sequences: "Forgery cake cooking" and "Forgery Awesome Cuponk" which both contain "fade" cut scenes. "Forgery cake cooking" also contains evidence of temporal upsampling from 25 fps to 30 fps. The robustness of this method of i-frame detection against temporal upsampling has not been investigated, and this is left for future work. Since spatially non-uniform temporal upsampling

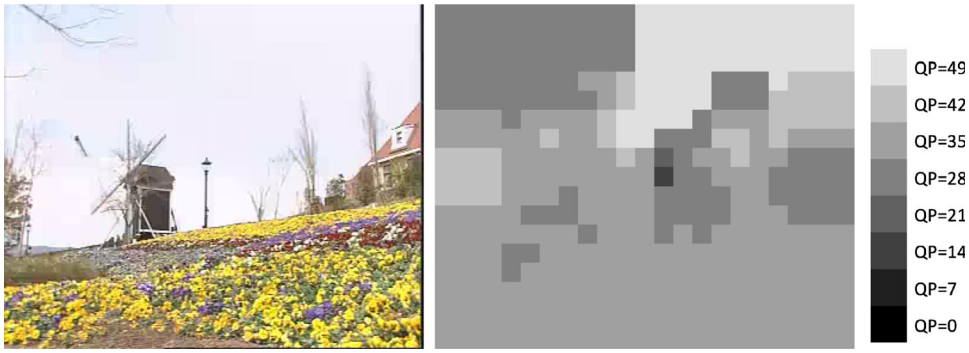


Fig. 3 QP heatmap for test sequence "flowers", QP = 35. The heatmap gives an average prediction of QP = 35, but there is some variation between individual regions

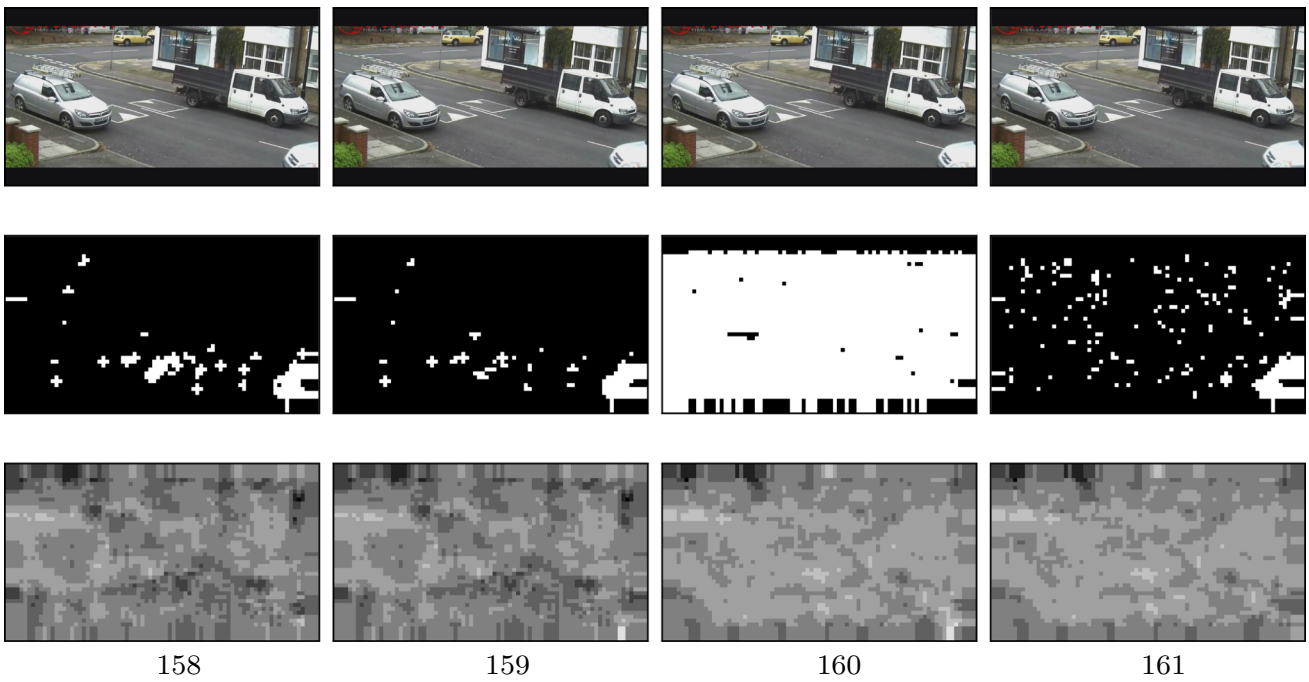


Fig. 4 Frames 158–161 of sequence "forgery CCTV_London_Str" [20], showing (top to bottom) sequence, binary frame difference for 16 × 16 blocks (black = no difference, white = differences) and QP prediction using a trained neural network. Frame differences clearly indicate the key frame, even though it is not visible in the sequence. The key frame is frame 160

would be indicative of video splicing, a method to detect it would prove useful.

The i-frame detection method was also applied to D'Avino et al's dataset [19]. Although the dataset is supplied as uncompressed .avi files, the i-frame detector provided evidence of previous compression by locating regular key frames at approximately 30 frame intervals.

4.3 Tampering detection analysis

In order to first show that predicted compression parameters can be used to locate tampering in video frames, we first examine the profiles of the different datasets. Table 4 shows predicted QP, averaged over the regions defined by the binary tampered mask. The last column in Table 4 shows the absolute difference in average QP per sequence averaged over all sequences. It can be seen that there is a distinct difference in QP averaged over authentic and tampered regions, particularly for the spliced content of [19] and the digitally manipulated content of [4], where the average absolute difference is larger than the granularity of the QP classifier. Figure 8a shows the predicted QP class distribution for sequence "08_TREE" [19], showing that authentic regions and spliced regions display different quantisation parameter distributions. The tampered content, in this instance, has a lower QP while the authentic content displays higher QP. The sequences of [19] consist of authentic content filmed on hand-held camera phones and green screen plates.[4] It can be deduced that, for this sequence, the hand-held cameras produced video of a lower quality than the green screen plates, resulting in distinct

[4] Some spliced content of [19] came from https://www.hollywoodcamerawork.com/green-screen-plates.html.

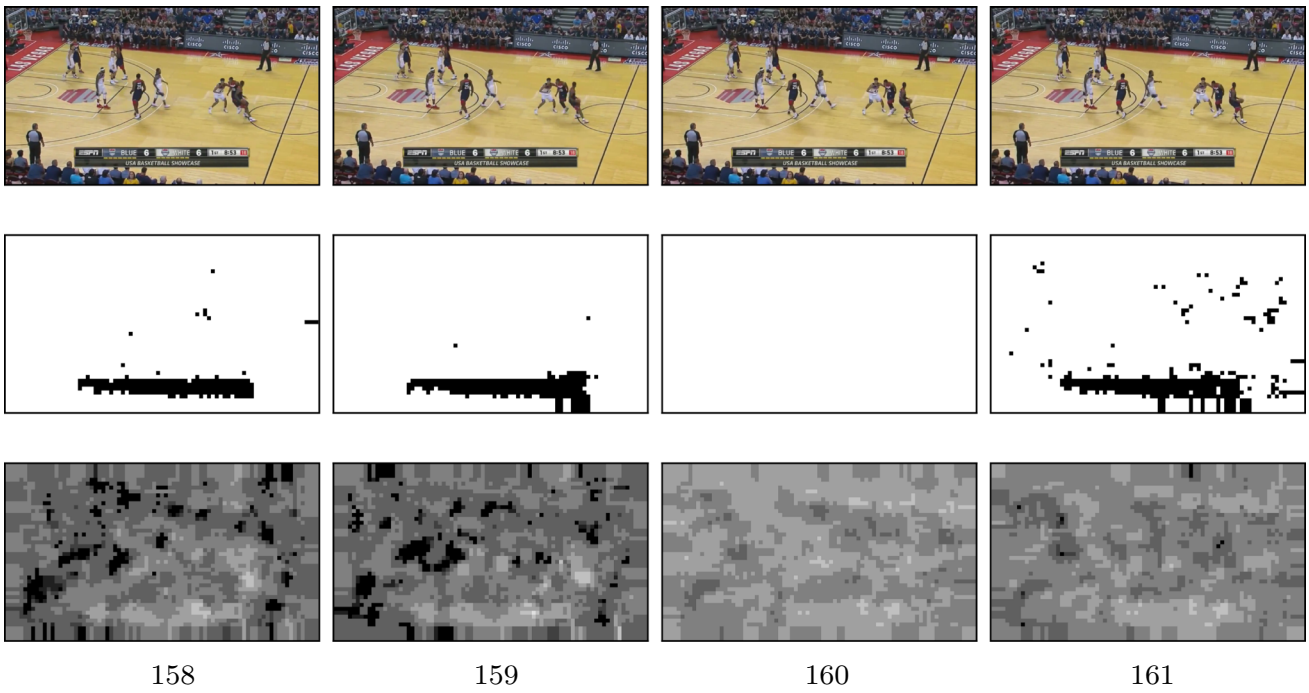


Fig. 5 Frames 158–161 of sequence "forgery basketball skills" [20], showing (top to bottom) original sequence, binary frame difference for 16 × 16 blocks (black = no difference, white = differences) and QP prediction using a trained neural network. Frame differences do not always highlight key frames. The key frame is frame 160

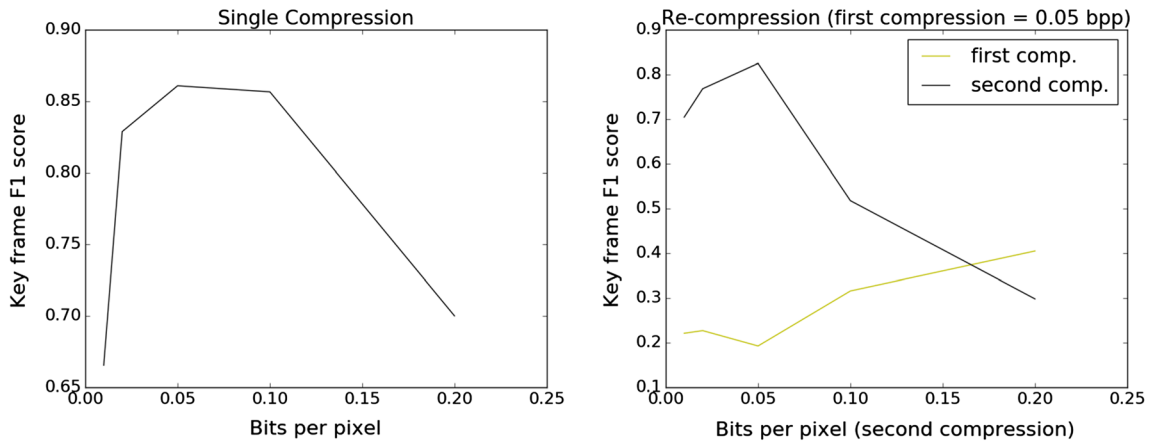


Fig. 6 F1 scores for key frame identification in singly and doubly compressed sequences

differences in QP distribution. Figure 8b shows the QP distributions over the first frames of FaceForensics, with the tampered points upsampled to equalise the dataset imbalance. It can be seen that, in general, tampered regions are compressed more lightly than authentic ones. Figure 9 demonstrates this further by displaying the QP heatmaps over some examples. It can be clearly seen that the tampered facial regions display lower QP than the authentic regions.

The copy–move content of VTD does not display a marked difference in predicted QP parameters because all copy–move content comes from within the same sequence and hence same QP distribution as shown in Fig. 8c. The difference for spliced content of VTD is slightly higher, but not significant enough for our CNN QP predictor to ascertain which distribution individual regions come from. The training set for our QP predictor used QP steps of 7, and the difference between spliced and authentic content of VTD is smaller than this. This effect may be due to the recompression step in the processing of this video: if the quality of both spliced and authentic content was reduced during recompression, then any differences in QP distribution will be consequently smoothed.

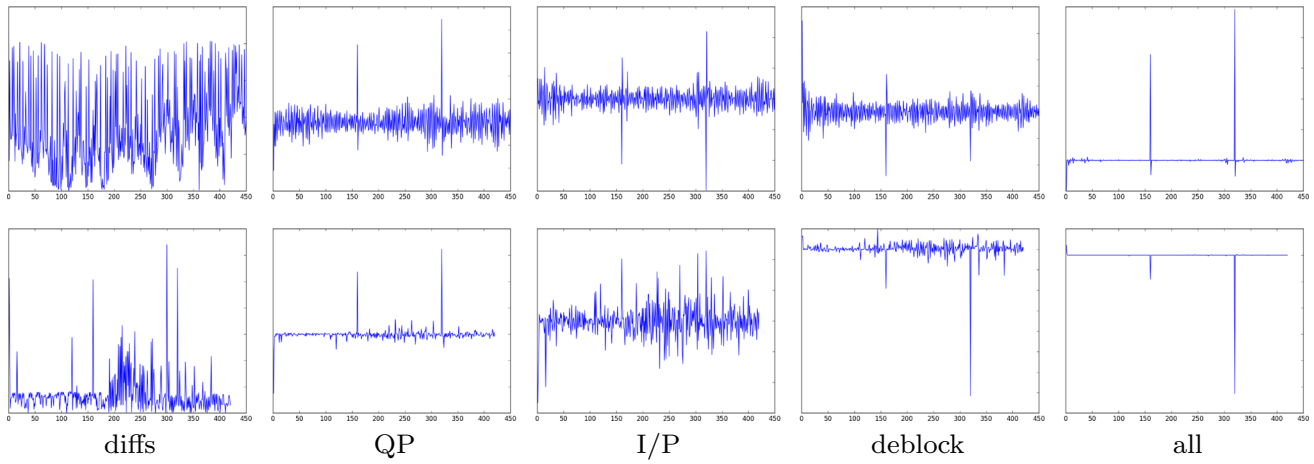
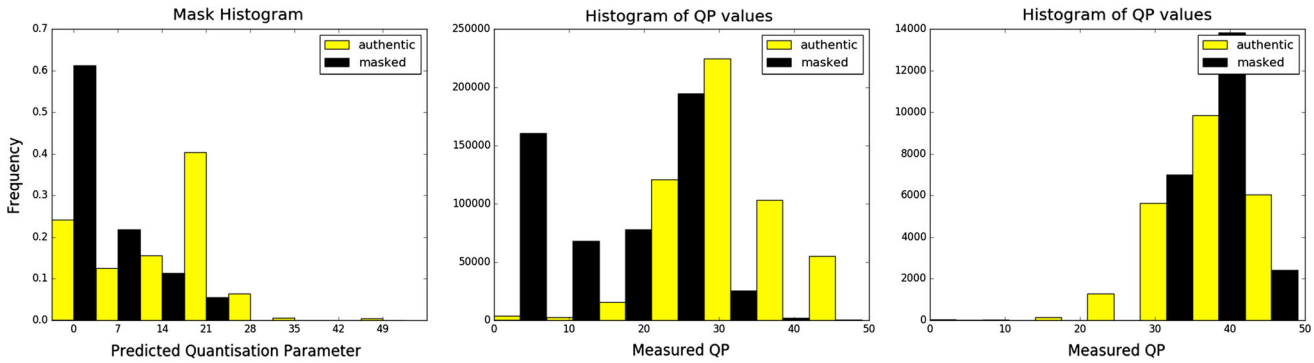


Fig. 7 Graphs showing differences between the mean value per frame of each feature for the sequences basketball (top) and cctv [20]. Frame averages versus frame number for (left to right): absolute frame differences; predicted QP; predicted inter/intra; predicted deblock; combination as in Eq. 3. The key frames can be clearly identified as outliers using a combination of the CNN predicted features

Table 4 Predicted QP on authentic and tampered pixels in detected key frames

Sequences	Average QP (mask = 0)	Average QP (mask = 1)	Average absolute diff.
FaceForensics [4] (Face2Face)	26.81	11.27	15.54
D'Avino [19] (splice)	19.12	9.44	9.68
VTD [20] (copy–move)	33.26	34.03	3.67
VTD [20] (splice)	32.58	26.98	5.80



(a) Splice "08_TREE" **(b)** FaceForensics **(c)** "dahua"

Fig. 8 Predicted QP class distribution for authentic and spliced content in sequence "08_TREE" [19], digital manipulation in FaceForensics [4] and copy–move content in sequence "dahua" [20]

Table 5 shows frame deltas (Sect. 3.6) averaged over regions and sequences. It can be seen that the averaged frame deltas for the tampered and authentic content of [19] are very close. There is a much bigger difference in VTD's spliced content. This is because some sequences, such as "Forgery Billiards" and "Forgery Studio", simply used static images as their spliced content. Similarly, some of the copy–move sequences, such as "Forgery basketball skills" and "Forgery 100m swimming", also used static content; however, since the tampered areas are also relatively static, it is not clear if this is an explicit feature of the tampering itself or simply of the region that was tampered.

4.4 Tampering localisation

Using all three compression parameters from the cropped FaceForensics test patches and unsupervised k-means clustering, assuming two clusters, we achieved MCC of 0.67 and mean F1 score of 0.81. Clustering could be validly

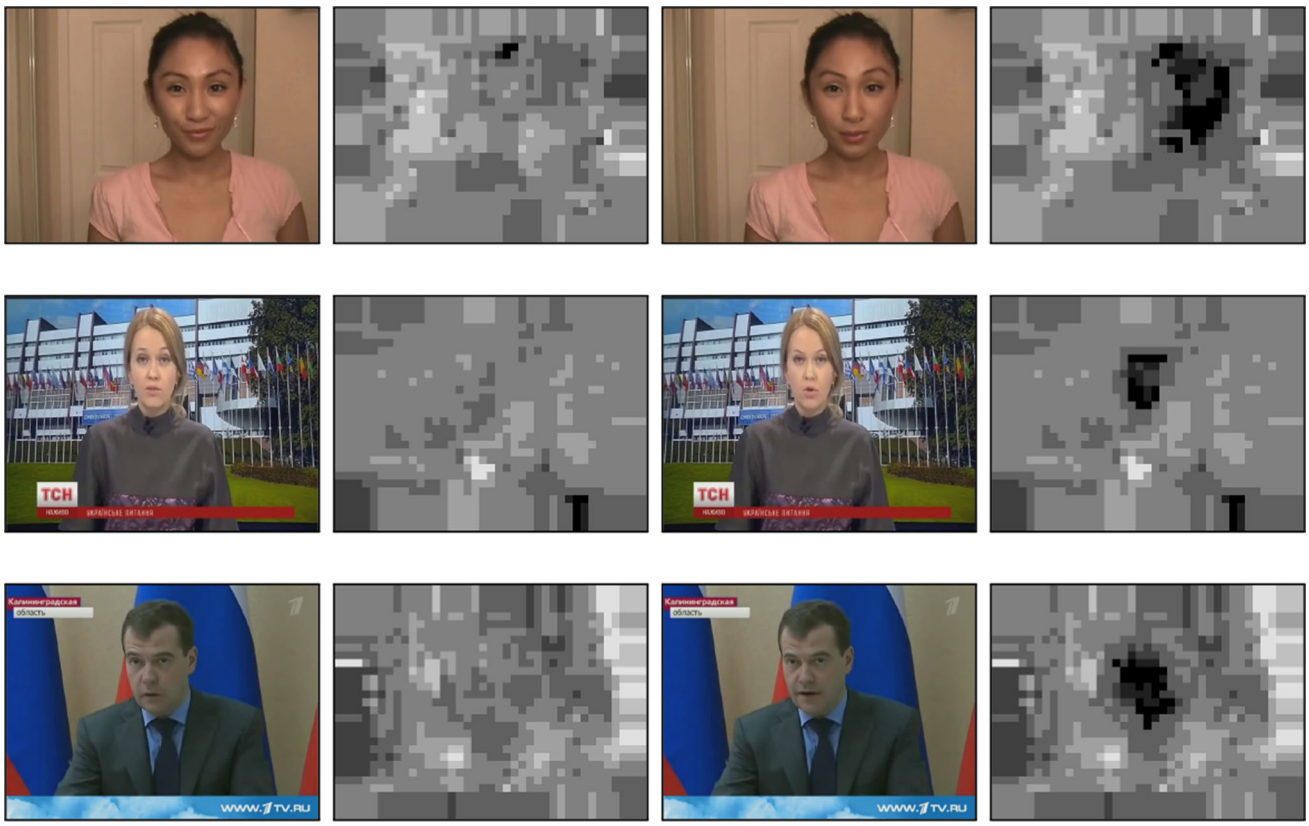


Fig. 9 Predicted QP is lower in the tampered regions of samples from FaceForensics [4], left to right: authentic pixels, authentic QP heatmap, tampered pixels, tampered QP heatmap

Table 5 Predicted frame deltas on authentic and tampered pixels in detected key frames (not applicable to FaceForensics datasets)

Sequences	Average diff (mask = 0)	Average diff (mask = 1)	Average absolute diff.
VTD [20] (copy–move)	0.68	0.43	0.31
VTD [20] (splice)	0.72	0.56	0.32
D'Avino [19] (splice)	0.90	0.92	0.11

performed over the whole dataset because the tampering method is the same for all sequences, and the source video footage all came from a single platform (YouTube). Unfortunately k-means clustering did not perform as well with full first frames of FaceForensics, achieving MCC = 0.11, but when this dataset was balanced using random oversampling, MCC rose to 0.58 with mean F1 over both classes of 0.76. This clearly demonstrates that there are dataset imbalance challenges associated with tampering detection.

Using only frame deltas and predicted QP as features and forming two clusters using unsupervised k-means clustering for each sequence, the following were achieved on D'Avino et al's dataset [19]: mean MCC: 0.249, mean F1 score: 0.255. MCC rises to 0.302 if the two lowest scoring sequences ("07_UFO" and "03_Cat") are ignored. In both these sequences, spliced objects are small relative to the frame size and occupy few key frames. Moreover, QP distributions for authentic and tampered regions in these two sequences are much less distinct that in the other sequences, and fall under the QP step size of 7 necessary for our CNN QP predictor.

In all the sequences of [19], key frames were estimated to occur at an interval of one every 30 frames. While this dataset is supplied as uncompressed, it is evidently compiled from compressed sources and one key frame per second at a frame rate of 30 fps is relatively standard in compression. Figure 10 shows example frames for some of the sequences from [19]. It can be seen that, for these examples, the quantisation parameters in tampered regions are generally lower than in authentic content. The results of the unsupervised k-means clustering, although noisy, reflect this. It can also be seen that there are a high number of false positives, and this combined with the dataset imbalance contributes to the relatively low MCC score.

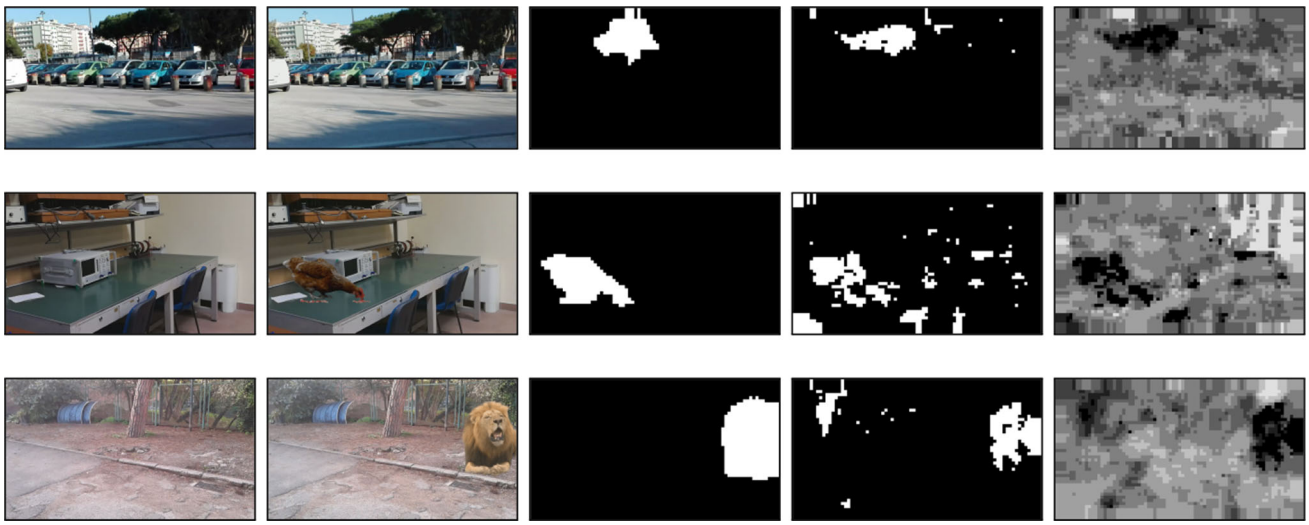


Fig. 10 Heatmap for test sequence (top to bottom) "08_TREE", "05_HEN" and "06_LION" from [19]: (left to right) real, fake, ground truth, clustered data, QP predictions. Darker areas mean lower QP predictions

The results on VTD [20] were somewhat less encouraging. There are relatively few key frames in many of these sequences. Where key frames were detected every 30 frames in [19], the most common gap between key frames in VTD was 160 frames, resulting in only 3 key frames for over half of the sequences in the dataset. Three sequences completely lacked key frames coinciding with the tampered region, and these were removed from our analysis. The use of MCC and F1 scores based on spatial components is unsuitable for frame shuffling. Excluding these sequences, the mean MCC was 0.082 and F1 0.065. This shows a weak correlation between predicted and actual tampered areas, which can be partly attributed to recompression causing an equalisation in the QP distribution between tampered and authentic regions. Although a realistic process, recompression of this dataset also resulted in challenges in extracting accurate tampering masks and this may also be a contributory factor. The authors of [19] also noted that YouTube compression had a negative effect on their auto-encoder-based tampering detector. This highlights challenges for tampering detection in video distributed using one of the most common video sharing platforms in the world. Further work is needed if tampering detectors are to thoroughly overcome the challenges of recompression.

5 Conclusions and future work

With video manipulation techniques currently increasing at an unprecedented rate, it is vital to develop features that can detect tampering irrespective of the original tampering method. A lack of large, current, comprehensive tampered video datasets makes learning these features from tampered data impossible; therefore, it is necessary to derive such features using authentic sources. Video compression provides a common foundation for video analysis, with the vast majority of available video sequences compressed in some format. Moreover, the use of machine learning techniques and feature discovery from data provides a methodology which can be used to produce updated features should new compression standards fall in to common use.

We have shown that three features of H.264/AVC compression, namely quantisation parameter, intra/inter and deblock modes, can be estimated objectively by CNN. These features have been used to predict the location of key frames in a video sequence, where they provide some advantage over simple frame deltas. They have also been used to localise spliced regions within the detected key frames. Results suggest that this type of feature shows great promise in the work towards universal tampering detection. Video manipulation causes self-inconsistencies within the video sequence, whether this is caused by splicing, inpainting, inter-frame tampering or small, localised changes used to alter content such as those used in digital re-enactment. This work shows that with the use of only four features (QP, inter/intra, deblocking and frame differences) derived exclusively from untampered sources, self-inconsistencies within a video sequence can be detected and exploited to localise tampering.

Our future work will examine further features that can be learned from authentic video and used to refine the localisation of video manipulation. A finer grained quantisation parameter predictor would improve prediction with the current feature set; however, this might require

migration from spatial to frequency domain. Additional features to investigate include compression features, particularly those associated with non-key frames such as skipped macroblocks and motion vector regions and features specific to multiple compressions. Other processing steps performed by cameras or software in the video processing pipeline should also be examined to determine if these are robust against recompression. We will also work on a better method to combine these features into a video manipulation localiser which is robust against multiple types of tampering.

Compliance with ethical standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Liu M-Y, Breuel T, Kautz J (2017) Unsupervised image-to-image translation networks. In: Advances in neural information processing systems, pp 700–708
2. Suwajanakorn S, Seitz SM, Kemelmacher-Shlizerman I (2017) Synthesizing obama: learning lip sync from audio. ACM Trans Graph 36(4):95
3. Thies J, Zollhöfer M, Stamminger M, Theobalt C, Nießner M (2016) Face2face: real-time face capture and reenactment of RGB videos. In: IEEE conference on computer vision and pattern recognition (CVPR), 2016. IEEE, pp 2387–2395
4. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M (2018) Faceforensics: a large-scale video dataset for forgery detection in human faces. arXiv preprint arXiv:1803.09179
5. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M (2019) Faceforensics++: learning to detect manipulated facial images. arXiv preprint arXiv:1901.08971
6. ITU-T. H.264 Advanced video coding for generic audiovisual services. ITU-T, 10 2016
7. ITU-T. H.262 Information technology—Generic coding of moving pictures and associated audio information: Video. ITU-T, 2 2012
8. Dodge S, Karam L (2016) Understanding how image quality affects deep neural networks. In: Eighth international conference on quality of multimedia experience (QoMEX), 2016, pp 1–6
9. Johnston P, Elyan E, Jayne C (2018) Spatial effects of video compression on classification in convolutional neural networks. In: International joint conference on neural networks (IJCNN), 2018, pp 1370–1377
10. Johnston P, Elyan E (2019) A review of digital video tampering: from simple editing to full synthesis. Digit Investig 29:67–81
11. Sitara K, Mehtre BM (2016) Digital video tampering detection: an overview of passive techniques. Digit Investig 18:8–22
12. Qureshi MA, Deriche M (2015) A bibliography of pixel-based blind image forgery detection techniques. Signal Process Image Commun 39:46–74
13. Chauhan D, Kasat D, Jain S, Thakare V (2016) Survey on keypoint based copy-move forgery detection methods on image. Procedia Comput Sci 85:206–212
14. Pandey RC, Singh SK, Shukla KK (2016) Passive forensics in image and video using noise features: a review. Digit Investig 19:1–28
15. Singh RD, Aggarwal N (2017) Video content authentication techniques: a comprehensive survey. Multimedia Syst 24:1–30
16. Ravi H, Subramanyam AV, Gupta G, Avinash KB (2014) Compression noise based video forgery detection. In: IEEE international conference on image processing (ICIP), 2014, pp 5352–5356
17. Lin CS, Tsay JJ (2014) A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis. Digit Investig 11(2):120–140
18. Johnston P, Elyan E, Jayne C (2018) Toward video tampering exposure: inferring compression parameters from pixels. In: Elias P, Chrisina J (eds) Engineering applications of neural networks. Springer International Publishing, pp 44–57
19. D'Avino D, Cozzolino D, Poggi G, Verdoliva L (2017) Autoencoder with recurrent neural networks for video forgery detection. Electron Imaging 7:92–99
20. Al-Sanjary OI, Ahmed AA, Sulong G (2016) Development of a video tampering dataset for forensic investigation. Forensic Sci Int 266:565–572
21. Khodabakhsh A, Ramachandra R, Raja K, Wasnik P, Busch C (2018) Fake face detection methods: can they be generalized? In: 2018 international conference of the biometrics special interest group (BIOSIG). IEEE, pp 1–6
22. Agarwal S, Fan W, Farid H (2018) A diverse large-scale dataset for evaluating rebroadcast attacks. In: IEEE international conference on acoustics, speech, and signal processing
23. Huh M, Liu A, Owens A, Efros AA (2018) Fighting fake news: image splice detection via learned self-consistency. In: The European conference on computer vision (ECCV)
24. Qadir G, Yahaya S, Ho ATS (2012) Surrey university library for forensic analysis (sulfa) of video content
25. Shullani D, Fontani M, Iuliani M, Al Shaya O, Piva A (2017) Vision: a video and image dataset for source identification. EURASIP J Inf Secur 1:15
26. Shanableh T (2013) Detection of frame deletion for digital video forensics. Digit Investig 10(4):350–360
27. Bosse S, Maniry D, Wiegand T, Samek W (2016) A deep neural network for image quality assessment. In: IEEE international conference on image processing (ICIP). IEEE, pp 3773–3777
28. Chen Y-J, Lin Y-J, Hsieh SL (2016) Analysis of video quality variation with different bit rates of h. 264 compression. J Comput Commun 4(05):32
29. Richardson Iain E (2011) The H. 264 advanced video compression standard. Wiley, New York
30. ITU-T. H.265 High efficiency video coding. ITU-T, 12 2016
31. Schaefer G, Stich M (2003) UCID: an uncompressed color image database. In: Storage and retrieval methods and applications for multimedia 2004, vol 5307. International Society for Optics and Photonics, pp 472–481
32. Stuart L (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137
33. Rao Y, Ni J (2016) A deep learning approach to detection of splicing and copy-move forgeries in images. In: IEEE international workshop on information forensics and security (WIFS), 2016. IEEE, pp 1–6

34. Credits for the use of the casia image tempering detection evaluation database (casia tide) v2.0 are given to the national laboratory of pattern recognition, institute of automation, chinese academy of science, corel image database and the photographers. http://forensics.idealtest.org
35. Sutthiwan P, Shi YQ, Zhao H, Ng T-T, Su W (2011) Markovian rake transform for digital image tampering detection. In: Transactions on data hiding and multimedia security VI. Springer, pp 1–17
36. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980