AUTHOR(S): JOHNSTON, P., ELYAN, E. and JAYNE, C.

TITLE: Toward video tampering exposure: inferring compression parameters from pixels.

YEAR: 2018

Publisher citation: JOHNSTON, P., ELYAN, E. and JAYNE, C. 2018. Toward video tampering exposure: inferring compression parameters from pixels. In Pimenidis, E. and Jayne, C. (eds.) Communications in computers and information science, 893: engineering applications of neural networks; proceedings of the 19th international engineering applications of neural networks conference (EANN 2018), 3-5 September 2018, Bristol, UK. Cham: Springer [online], pages 44-57. Available from: https://doi.org/10.1007/978-3-319-98204-5_4

OpenAIR citation: JOHNSTON, P., ELYAN, E. and JAYNE, C. 2018. Toward video tampering exposure: inferring compression parameters from pixels. In Pimenidis, E. and Jayne, C. (eds.) Communications in computers and information science, 893: engineering applications of neural networks; proceedings of the 19th international engineering applications of neural networks conference (EANN 2018), 3-5 September 2018, Bristol, UK. Cham: Springer, pages 44-57. Held on OpenAIR [online]. Available from: https://openair.rgu.ac.uk/

A Convolutional Neural Network to Measure H.264/AVC Compression

Pamela Johnston[1], Eyad Elyan[1], and Chrisina Jayne[2]

[1] Robert Gordon University, Aberdeen, United Kingdom
[2] Oxford-Brookes University, Oxford, United Kingdom

Abstract. Video tampering detection remains an open problem in the field of digital media forensics. Some existing methods focus on recompression detection because any changes made to the pixels of a video will require recompression of the complete stream. Recompression can be ascertained whenever there is a mismatch between the compression parameters encoded in the syntax elements within the compressed bitstream and those derived from the pixels themselves. However, deriving compression parameters directly and solely from the pixels is not trivial. In this paper we propose a new method to estimate the H.264/AVC quantisation parameter (QP) in frame patches from raw pixels using Convolutional Neural Networks (CNN) and class composition. Extensive experiments show that QP of key-frames can be estimated using CNN. Results also show that accuracy drops for predicted frames. These results open new interesting research direction in the domain of video tampering/forgery detection.

Keywords: CNN · Compression · video tampering detection

1 Introduction

In the age of fake news and falsified video, the detection of video tampering is becoming an increasingly important area of research. Techniques such as [1, 2] demonstrate how the latest machine learning techniques can convincingly alter video content by changing faces or weather conditions, yet detection of such tampering remains an open field. Detection methods can be active or passive [3, 4], but, since many existing videos are unprepared for active tampering detection, passive detection methods are more relevant. Passive tampering detection can be categorised into recompression, region tampering and inter-frame forgery [3]. Region tampering includes copy-move attacks where the copied region can come from the same frame in the video, similar to image copy-move [5] or from a different frame in the same video [6]. Splicing and inpainting are variations on region tampering. Inter-frame forgery is where an integer number of frames is added, deleted or shuffled. Regardless of the editing method, however, any tampering at the pixel level of a compressed video requires recompression of the video bitstream [7, 8], so of these three methods, recompression detection is the most versatile.

Video compression is prevalent in digital society. The vast majority of online video has been compressed using lossy formats such as H.264/AVC [9] or MPEG2 [10]. These formats have been designed with the human visual system in mind and the effects of compression remain largely invisible to human eyes. It has been shown that compression does impact classification performance of convolutional neural network (CNN) classifiers [11] and pre-existing compression in original source images may even have caused these effects to be understated. If CNN classifiers are passively affected by compression, it is reasonable to use them to actively detect the level of compression directly from pixels. Moreover, any method of measuring compression could be utilised to create an ensemble CNN classifier which could maximise accuracy while accounting for recompression. Accurate QP estimation could be used to enhance the performance of classifiers across differing quality levels.

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. The human visual system cannot distinguish between close QP levels, however and objective methods of measuring QP from pixels are required. An ideal QP estimator would also operate accurately over small patches to enable localisation of tampered regions because this is an advancing area of research [12, 4]. 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 takes a step towards utilising compression parameters derived directly from the pixels themselves. We show CNNs can be trained to estimate QP for stand alone key frame patches with reasonable accuracy. Original datasets are synthesised from uncompressed sources and used to train CNNs to identify the QP used to encode the data. We train a CNN to estimate the quantisation parameter of a pixel patch singly encoded using H.264/AVC. The accuracy of our model is examined and contributory factors to errors, including the reasons for lower accuracy on predicted frames, are explored. Class composition is also used to improve accuracy in predicted frames. Unlike [13], where decomposition of a original dataset classes into *smaller* subclasses improved accuracy in random forest classification, we combine adjacent classes into *larger* superclasses. We find that training a CNN on superclasses improves accuracy. We explain how our model works through examination of the network weights.

2 Background and related work

The human visual system is adequate to detect some compression effects and can quantify "no reference" image and video quality [14, 15]. The source of video compression visual effects can be found by examining transformations used in compression standards. A video sequence comprises key or intra frames, which provide access points into the sequence, and predicted frames which rely on data from previously encoded frames. In H.264/AVC and MPEG-2, frames are divided into "macroblocks": blocks of 16x16 pixels. For non-predicted data, the pixel data itself is transformed into the frequency domain using Discrete Cosine Transforms (DCT), 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 Equation 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 [16].

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

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 [10] where non-integer DCTs forced regular inclusion of key frames. Periodic key frames limited drift between encoder and decoder but were visible as a pulse in the sequence as accumulated rounding errors were reset by the key frame. The integer transforms introduced in H.264/AVC [9] reduced the role of key frames to access points in the bitstream and consequently reduced the visible pulse in video sequences. HEVC [17] 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.

Traditional methods of recompression detection rely on the identification of patterns in frequency domain bitstream syntax elements. The authors of [18] rely on Benford's distribution of DCT coefficients and support vector machines to detect double compression of intra frames. In [19] multiple compression is detected in H.264/AVC encoded videos but the compression modes are heavily restricted and the methods do not differentiate between QP that are less than two steps apart.

As part of an investigation into using deep neural networks to determine image quality, Bosse et al [14] 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 [20]. The method was applied to intra (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 the averaged QP prediction for a complete frame was accurate, a heatmap showing individual patch contributions displayed great variation between patches. If QP estimation is to be successful as a region-tampering detector, it should be as accurate as possible over small regions. Moreover, a QP estimator for video must also handle *predicted* frames.

This work examines QP estimation in the context of patches taken from H.264/AVC video sequences. 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 [14], we also investigate predicted frames in a video sequence.

3 Methods

3.1 Datasets

When examining the effects of compression, is vital to start with unprocessed data. Standard YUV 4:2:0 sequences from xiph.org are commonly used for video compression quality analysis[3]. Strictly speaking, YUV 4:2:0 is a compressed format due to reduced resolution of the colour channels but it is widely used in video compression. Uncompressed YUV 4:4:4, is not as popular. 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. Camera motion varies between sequences but frames from a single sequence will be correlated.

A large amount of data is 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 [20] were used. UCID consists of uncompressed images which are either 512x384 pixels or 384x512 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 for download[4]. 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 and encoded accordingly as an intra frame.

Table 1. A summary of original datasets

Name	Source	Length	Dimensions	Key frames
CIFvid	xiph.org	18 videos	352x288	1/250
CIFintra	xiph.org	18 videos	352x288	all
AllVid	xiph.org	44 videos	176x144 to 1920x1080	1/250
AllIntra	xiph.org	44 videos	176x144 to 1920x1080	all
UCID	UCID [20]	882 single frames	512x384 or 384x512 pixels	all

Table 1 gives a summary of the original datasets. Each video sequence was compressed using the open source H.264/AVC encoder x264 and one of a range

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

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

Table 2. A summary of synthesised datasets

Name	Source	Patch Size	Spatial Stride	Temp. Stride	Train Patches	Test Patches
AllVid_80	AllVid	80	80(train); 40(test)	40	156592	8400
AllIntra_80	AllVid	80	80(train); 40(test)	40	156592	8400
UCID_80	UCID	80	80	1	131904	53480
CIFvid_80	CIFvid	80	48	30	79920	7920
AllVid_32	AllVid	32	80(train); 40(test)	40	191776	13872
AllIntra_32	AllVid	32	80(train); 40(test)	40	191776	13872
UCID_32	UCID	32	80	1	183320	26320
UCID_32_large	UCID	32	32	1	974528	140512
CIFvid_32	CIFvid	32	32	60	118976	12672
CIFintra_32	CIFvid	32	32	60	118976	12672

of constant QP levels using variable bitrate mode. Constant quantisation parameters were selected with an even distribution: QP=[0, 7, 14, 21, 28, 35, 42, 49]. Constant bitrate rate control, psychovisual options and deblocking filter were turned off. For datasets containing predicted frames, the key frame interval was 250. Patches were then extracted from the decoded YUV4:2:0 sequences. Patches were converted from YUV4:2:0 to YUV4:4:4, where the Y-channel represents intensity and U and V channels are colour. Table 2 summarises synthesised datasets. A large temporal stride was used to limit correlation between patches. Consecutive frames are similar to each other and training a neural network with a correlated dataset will cause overfitting. Each patch was labelled with its quantisation parameter. All datasets were prepared in advance of network training and the original video sequences were split into train and test sets prior to compression and patch sampling to prevent data leakage[5].

Two different patch sizes were selected to investigate which aspects of compression were important to CNNs. Block edge artifacts in intra frames will present themselves at macroblock (and subblock) boundaries. Therefore, any patch size larger than 16x16 will capture block edge artifacts. Following [14], a small patch size of 32x32 was selected. A larger patch size of 80x80 pixels was also used. When aligned with the macroblock grid, 80x80 pixels covers 5x5 complete macroblocks. A larger patch size allows for more context and image features within the patch to contribute towards QP estimation. 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.

3.2 Network architectures

For the purposes of this paper, three simple network architectures (NAs) were examined, summarised in Table 3. Image patches were format YUV 4:4:4, rescaled to values between 0 and 1 and whitened. In order to preserve compression artifacts in situ, no further data augmentation was used. Batch size was 128 patches. Unless otherwise noted in the results, NAs 1 and 2 were implemented with stride = 2 for all convolutional and pooling layers.

Network architectures were designed with compression artifacts in mind. H.264/AVC uses a minimum DCT block size of 4x4 pixels so a 4x4 kernel aligns

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

Table 3. A summary of network architectures

Name	Layers
NA 1	conv4x4-64, pool3x3, norm, conv4x4-64, norm, pool3x3, fc-384, fc-192, softmax
NA 2	conv5x5-64, pool3x3, norm, conv5x5-64, norm, pool3x3, fc-384, fc-192, softmax
NA 3 [14]	conv3x3-32, conv3x3-32, pool2x2, conv3x3-64, conv3x3-64, pool2x2, conv3x3-128, conv3x3-128, pool2x2, conv3x3-256, conv3x3-256, pool2x2, conv3x3-512, conv3x3-512, pool2x2, fc-512, softmax

to this. Using an even-sized kernel is unusual but not without precedent [21]. A stride of 2 allows sufficient overlap to encounter artifacts while reducing the number of network parameters. Networks were trained and tested multiple times and average accuracy and confusion matrix values taken for the results.

3.3 Estimating quantisation accuracy

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

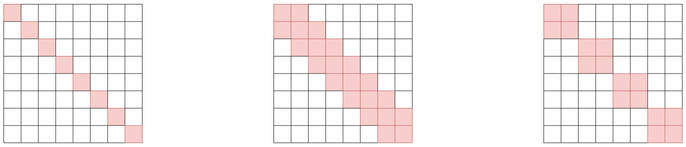
$$0 \leq QP \leq 52,\ QP \in \mathbb{R} \tag{2}$$

QP relates directly to Q_s in Equation 1. Patches with similar QP labels exhibit similar compression features, and confusion matrices produced by the network reflected this. Two different QPs might have very similar effects on a given patch, depending on the patch content. An example of this is a whole patch of solid colour, which transforms to a single high amplitude, low frequency coefficient which is non-zero 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 has been sampled at [0, 7, 14, 21, 28, 35, 42, 49] in the synthesised datasets. Using all possible QP would also generate an extremely large dataset and increase model training times. Using a range of sampled QP, the confusion matrices produced by the model can be examined and super-classes composed to estimate accuracy.

Unlike the work presented in [13], where data was decomposed based on discrete classes, in this paper, we combine each two adjacent classes into one, assuming that the change in pixels is not significant within adjacent QP. Figure 1 gives a visual demonstration of how this can be applied to a confusion matrix. In a confusion matrix, predicted labels from the network are tabulated against ground truth class labels. The overall accuracy of the network is given as the sum of the diagonal elements of the confusion matrix divided by the sum of all the elements (Fig. 1a). That is, the *correctly predicted* elements divided by all the elements. If some degree of error is permitted in the confusion matrix, adjacent classes can be accepted as "correct" and a new accuracy shown $\hat{p}$ (Fig. 1b) can be calculated by combining adjacent classes thus:

$$\hat{p} = \frac{1}{M}\{\sum_{i=0}^{m} a_{i,i} + \sum_{i=1}^{m} a_{i-1,i} + \sum_{i=1}^{m} a_{i+1,i} + \sum_{i=1}^{m} a_{i,i-1} + \sum_{i=1}^{m} a_{i,i+1}\} \tag{3}$$

Although Equation 3 combines adjacent classes well in theory, testing how well it represents a model where classes are combined *before* training would involve assigning different labels to identical data. Instead, we compose super classes according to Equation 4 and Figure 1c.



(a) Standard accuracy

(b) Combine adjacent classes

(c) Removing label ambiguity

Fig. 1. Different class compositions in a confusion matrix

$$\hat{p} = \frac{1}{M}\{\sum_{i=0}^{m/2} a_{2i,2i} + \sum_{i=0}^{m/2} a_{2i+1,2i} + \sum_{i=0}^{m/2} a_{2i,2i+1} + \sum_{i=0}^{m/2} a_{2i+1,2i+1}\} \tag{4}$$

Using Equation 4, labels can be unambiguously combined and a network trained on these labels. Equation 4 can also be extended to create even larger super-classes allowing for ever greater error.

4 Evaluation and discussion

The initial experiment using CIFVid_80 achieved only 36.25% accuracy. Following [14], patch size was reduced and UCID was introduced, creating two new datasets: CIFVid_32 and UCID_32_large. The results in Table 4 show that smaller patch size did not improve accuracy, but training on intra frames only did. Halving the network stride parameter helped, but was still worse than using a larger patch size. Networks trained on UCID_32_large achieved accuracy of over 58% when tested with UCID_32_large test data, but approximately half that when tested with CIFVid_32.

CIFintra_32, comprising all key frames (Table 2) answered the question of why learning from UCID_32_large did not translate well to CIFVid_32. Patches in CIFVid_32 and CIFintra_32 come from exactly the same points in the video sequences, so their content is strongly visually correlated. Only the underlying compression modes differ. The best performing network architecture was retested with CIFintra_32. The accuracy on CIFintra_32 using the network trained on UCID_32 showed good improvement over testing on CIFVid_32 (54.14% vs 30.35%).

CIFVid_32, CIFintra_32 and UCID_32_large were mismatched in terms of patch quantity within the training sets. To investigate whether this accounted for some of the differences in accuracy, AllVid_32, AllIntra_32 and UCID_32 were created. Table 5 shows the accuracy achieved on each combination. The larger training set UCID_32_large improved accuracy by an average 7.9% on the UCID_32 test set but only average 2.34% on AllVid_32. The addition of extra training video patches when increasing CIFVid_32 to AllVid_32 did not increase accuracy. From these differences in performance, it can be concluded that the UCID-based datasets were less correlated than those derived from video sequences leading to more feature coverage and more generalisable networks.

In Table 5 intra-only trained networks still out-performed those trained on AllVid_32, except in the case of AllIntra_32 versus AllVid_32. Given that AllVid_32 and AllIntra_32 contain visually similar pixel patches, this implies that the network trained on AllVid_32 has learned some features distinct to predicted frames that do not translate to key frames. This pattern was repeated with patch size 80x80.

Larger patch size datasets AllVid_80, AllIntra_80 and UCID_80 were generated and used to train and test different network architectures. Table 5 shows the results. Comparing results for AllVid_80 with CIFVid_80 and AllVid_32 (35.27%, 36.25%, 26.78%, respectively) show that increasing the number of patches did not improve accuracy but increasing the patch size did. Although networks trained and tested on UCID_80 achieved good accuracy (Table 5), the learning did not transfer to AllVid_80. It did, however, translate to AllIntra_80. From this, it can be deduced that QP can be successfully estimated directly from the pixels of key frames but does not translate well to predicted frames. Networks trained on predicted frame patches achieve lower accuracy than that those trained on key frame patches. Moreover, the accuracy of networks trained on UCID_80 is higher than those trained on AllIntra_80. This can be partly attributed to weaker correlation in UCID_80 image patches.

Overall, NA 3, the deepest network, had the lowest accuracy. The limited size of the datasets may have contributed to this, but it is more likely that the depth of the network did not help with compression features. Compression features in H.264/AVC are related to the size of the transforms used in the codec and these vary from 4x4 to 16x16 pixels. It can be deduced that though deeper networks go some way to accounting for differences in scale in traditional object classification, this is largely unnecessary when examining compression.

Table 4. Initial results: Accuracy for patch size 32 (network 3 failed to train).

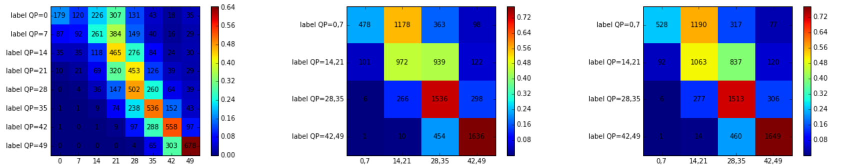
Network	Tested on	CIFVid_32 trained	UCID_32_large trained
1	CIFVid_32	27.30	33.14
1	UCID_32_large	31.00	58.55
2	CIFVid_32	26.69	30.35
2	UCID_32_large	32.23	59.28
2	CIFintra_32	27.30	54.14
2 (stride=1)	CIFVid_32	25.34	33.67
2 (stride=1)	UCID_32_large	28.46	66.88

4.1 Relaxing the problem

Although the overall accuracy achieved on a network trained on predicted frame patches from AllVid_80 was low, the confusion matrix implied a reasonable error rate. Figure 2a shows the confusion matrix for NA 2 trained/tested on AllVid_80. Average accuracy was 35.27%. With class labels combined as in equation 4, the accuracy estimated from the confusion matrix is 54.65%. A network trained on the reduced label dataset yields a comparable accuracy of 56.25%. Therefore, the results obtained from calculations on the confusion matrix after training are comparable to networks trained specifically on these super classes. Table 6 shows that this is true across all patch size 80 datasets. Composing super classes from adjacent classes prior to training reduces the number of labels and

Table 5. Cross evaluation on similar sized datasets. accuracy for patch size 32 (network 3 failed to train) and for patch size 80

	Patch size 32				Patch size 80			
Network	Tested on	AllVid_32 trained	AllIntra_32 trained	UCID_32 trained	Tested on	AllVid_80 trained	AllIntra_80 trained	UCID_80 trained
1	AllVid_32	26.72	24.68	31.34	AllVid_80	34.93	26.02	36.72
1	AllIntra_32	27.75	33.75	44.14	AllIntra_80	34.11	37.94	63.40
1	UCID_32	33.74	41.56	50.96	UCID_80	41.28	47.46	72.75
2	AllVid_32	26.78	25.38	31.79	AllVid_80	35.27	29.39	37.28
2	AllIntra_32	27.07	33.16	45.05	AllIntra_80	34.93	46.54	62.50
2	UCID_32	33.83	41.25	51.07	UCID_80	42.04	56.66	71.65
3	-	-	-	-	AllVid_80	29.97	24.91	29.35
3	-	-	-	-	AllIntra_80	29.94	42.67	55.95
3	-	-	-	-	UCID_80	38.99	53.81	61.17



(a) Full (35.27%) (b) Reduced (54.65%) (c) Composed (56.25%)

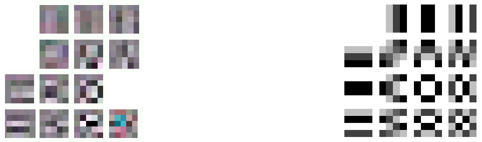
Fig. 2. Confusion matrices for NA 2 trained/tested on AllVid_80 (overall accuracy for a single network)

slightly enhances generalisation in the network, yielding slightly higher results from video-based datasets with correlation between video patches. The same pattern was repeated with other architectures, though results are omitted for space considerations. QP in predicted frames can be estimated to within ±7 (one class) with more than 54% accuracy. Higher quality frames are more challenging and this may be attributed to the larger range of frequencies available in uncompressed data. CNN models cannot distinguish between frequencies removed by compression and those simply absent in the source data.

Table 6. Accuracy for patch size 80 (NA 2): composition after/before training

Tested on	AllVid_80 trained	AllIntra_80 trained	UCID_80 trained
AllVid_80	54.65 / 56.25	52.99 / 54.08	60.70 / 59.55
AllIntra_80	55.24 / 56.51	66.62 / 67.55	78.81 / 78.47
UCID_80	66.81 / 67.94	78.58 / 79.57	88.43 / 88.41

The shape of the confusion matrices (Fig. 2) gives insight into the model's learning. Confusion matrices across all architectures and datasets demonstrated similar shapes where the bottom left corner approached zero. Patches of high QP were seldom misclassified as low QP. In contrast, the top right corner of the confusion matrix, although it displays lower numbers than the diagonal portion, does not always approach zero. This pattern suggests that the model has learned something about the frequency domain. Natural images contain a large variety of frequencies, however quantisation in the frequency domain selectively reduces these (as in Equation 1). Weaker (low amplitude) frequency components are quantised to zero and thus filtered out of an image, leaving behind only dominant frequencies. For natural scenes, where lower frequencies tend to dominate, quantisation applied in video compression is effectively a low pass filter. This explains the "blocky" appearance of compressed video. Low amplitude, high



(a) First layer filters (b) 4x4 DCT

Fig. 3. Some of the first layer filters display visual similarity to a spatial representation of the 4x4 DCT transform used in H.264/AVC

frequency components aid smooth colour or intensity transition and removing these components leads to sharper colour transitions at macroblock edges. The detection of high frequency components within macroblocks therefore indicates lower QP. Unfortunately, the converse is not necessarily true. An absence of high frequencies does not indicate high QP, since some images naturally lack high frequency components.

4.2 First layer filters

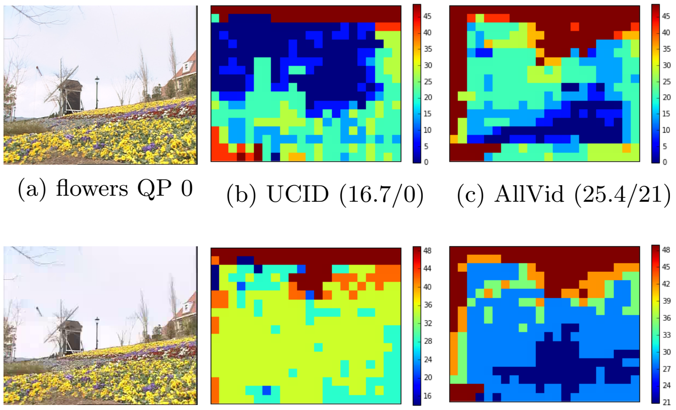
A visual examination of the first layer filters confirms the presence of frequency features. Figure 3b shows a spatial representation of the 4x4 Discrete Cosine Transform used in H.264/AVC. These are pixels that result from an inverse DCT on a 4x4 coefficient matrix where only one coefficient is non-zero. Figure 3a shows selected first layer filters from NA 2 trained on AllIntra_32. Visual similarities are obvious. The CNN uses some first layer filters to infer frequency information directly from the pixels. Statistical analysis of frequency was also used in [19] to detect multiple compression.

4.3 Whole image heat map

Figure 4 shows classification results for 80x80 patches of a key frame from the sequence "flowers" at QP 0 and 35. A plain black border was added added after compression to allow classification of 80x80 patches centred on every 16x16 macroblock. The heatmaps all show misclassification along the top row due to the black border. Comparing Figures 4a and 4d, it is difficult for human eyes to differentiate between QP values, despite the large difference. The fine colour transition in the sky section is correctly classified by a network trained on UCID_80 as low QP. The network trained on AllVid_80 tends to overestimate sky QP and perform better on the colourful flower section. At moderate QP, the model trained on UCID_80 performs well. The AllVid_80 trained network achieves a reasonable mean over the whole image but underestimates QP in busy areas and overestimates in areas with fewer sharp edges. Mode 28 in Figure 4f shows how the model confuses adjacent labels and validates the use of superclasses.

5 Conclusions and future work

We have shown that the level of compression of small image patches can be estimated objectively by CNN. The accuracy of CNNs trained on intra frames is



(a) flowers QP 0 (b) UCID (16.7/0) (c) AllVid (25.4/21)

(d) flowers QP 35 (e) UCID (35.6/35) (f) AllVid (33.0/28)

Fig. 4. The heat maps showing the predicted quant for a frame from the sequence "flowers", NA 2 used **(mean/mode)**

much higher than those trained on predicted frames. The experimental results also strongly suggest that compression features learned from still images (intra frames) alone do not transfer to predicted frames. Although a neural network can be trained to estimate the QP in key frame patches, the results for predicted frames were weaker. In predicted frames, quantisation is applied to the residual difference between predicted and actual pixels. CNN compression estimation may be improved by using residuals from the compressed bitstream rather than reconstructed pixels. It may also be necessary to first identify intra macroblocks within an image in order to gain an estimate of accuracy in QP estimation.

Larger patch sizes yield higher precision but further investigation will clarify whether an optimum patch size exists. Smaller patch sizes are desirable if the model is to serve as accurate tampering detection. The features learned in the first layer of CNNs strongly resemble patterns of DCT coefficients, so QP estimation may be improved by initialising some of the first layer weights with DCT patterns.

Compression is no longer an irreversible "black box". Although recompression destroys information about original compression mechanisms in the compressed bitstream, tell-tale signs in the pixels can be used to estimate original levels and this information could be used to detect video splicing as well as multiple compressions.

References

1. Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics*, 2017.
2. Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, 2017.

3. K Sitara and Babu M Mehtre. Digital video tampering detection: An overview of passive techniques. *Digital Investigation*, 2016.
4. Muhammad Ali Qureshi and Mohamed Deriche. A bibliography of pixel-based blind image forgery detection techniques. *Signal Processing: Image Communication*, 2015.
5. Devanshi Chauhan, Dipali Kasat, Sanjeev Jain, and Vilas Thakare. Survey on keypoint based copy-move forgery detection methods on image. *Procedia Computer Science*, 2016.
6. Ramesh C Pandey, Sanjay K Singh, and Kaushal K Shukla. Passive forensics in image and video using noise features: A review. *Digital Investigation*, 2016.
7. Raahat Devender Singh and Naveen Aggarwal. Video content authentication techniques: a comprehensive survey. *Multimedia Systems*, 2017.
8. Hareesh Ravi, AV Subramanyam, Gaurav Gupta, and B Avinash Kumar. Compression noise based video forgery detection. In *IEEE International Conference on Image Processing*. IEEE, 2014.
9. ITU-T. *H.264 Advanced video coding for generic audiovisual services*. ITU-T, 2016.
10. ITU-T. *H.262 Information technology - Generic coding of moving pictures and associated audio information: Video*. ITU-T, 2012.
11. Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *International Conference on Quality of Multimedia Experience*. IEEE, 2016.
12. Cheng-Shian Lin and Jyh-Jong Tsay. A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis. *Digital Investigation*, 2014.
13. Eyad Elyan and Mohamed Medhat Gaber. A fine-grained random forests using class decomposition: an application to medical diagnosis. *Neural computing and applications*, 2016.
14. Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. A deep neural network for image quality assessment. In *IEEE International Conference on Image Processing*. IEEE, 2016.
15. Yen-Jen Chen, Yang-Jen Lin, and Sheau-Ling Hsieh. Analysis of video quality variation with different bit rates of h. 264 compression. *Journal of Computer and Communications*, 2016.
16. Iain E Richardson. *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.
17. ITU-T. *H.265 High efficiency video coding*. ITU-T, 2016.
18. Tanfeng Sun, Wan Wang, and Xinghao Jiang. Exposing video forgeries by detecting mpeg double compression. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2012.
19. Simone Milani, Paolo Bestagini, Marco Tagliasacchi, and Stefano Tubaro. Multiple compression detection for video sequences. In *IEEE International Workshop on Multimedia Signal Processing*. IEEE, 2012.
20. Gerald Schaefer and Michal Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia*. International Society for Optics and Photonics, 2003.
21. Luca Bondi, Silvia Lameri, David Güera, Paolo Bestagini, Edward J Delp, and Stefano Tubaro. Tampering detection and localization through clustering of camera-based cnn features. In *CVPR Workshops*, 2017.