information

Article

A Survey on Compression Domain Image and Video Data Processing and Analysis Techniques

Yuhang Dong and W. David Pan *

Department of Electrical and Computer Engineering, University of Alabama in Huntsville, 301 Sparkman Dr, Huntsville, AL 35899, USA
* Correspondence: pand@uah.edu

Abstract: A tremendous amount of image and video data are being generated and shared in our daily lives. Image and video data are typically stored and transmitted in compressed form in order to reduce storage space and transmission time. The processing and analysis of compressed image and video data can greatly reduce input data size and eliminate the need for decompression and recompression, thereby achieving significant savings in memory and computation time. There exists a body of research on compression domain data processing and analysis. This survey focuses on the work related to image and video data. The papers cited are categorized based on their target applications, including image and video resizing and retrieval, information hiding and watermark embedding, image and video enhancement and segmentation, object and motion detection, as well as pattern classification, among several other applications. Key methods used for these applications are explained and discussed. Comparisons are drawn among similar approaches. We then point out possible directions of further research.

Keywords: compression domain; image; video; DCT; JPEG; MPEG; motion vector

Citation: Dong, Y.; Pan, W.D. A Survey on Compression Domain Image and Video Data Processing and Analysis Techniques. *Information* **2023**, *14*, 184. https://doi.org/10.3390/info14030184

Academic Editor: Heming Jia

Received: 23 December 2022
Revised: 8 March 2023
Accepted: 10 March 2023
Published: 15 March 2023

1. Introduction

It is estimated that 3.2 billion images and 720,000 h of video are shared online daily [1]. Image and video compression allows for the efficient storage and transmission of image and video data. Rather than simply compressing data streams to save space, the compression of images and video has also promoted development in other areas such as smart cities [2–4], healthcare [5–10] and agriculture [11].

Despite the relatively smaller size after compression, users still need to decompress data on the application side. If these data can be processed and analyzed directly in their compressed forms, then we can save storage space and computation time. The explanation is that in this case, we can work with significantly smaller input in the compression domain, eliminating the need to decompress the already compressed data for processing or analysis and then recompress the data back to their compressed form. In the literature, there exists a body of work which aims to investigate the feasibility and effectiveness of data analysis either partially or fully in the compression domain. By conducting analyses, e.g., object detection, on data in a compressed form, the input will be compressed data instead of original data. The obvious advantages of this approach include faster processing and analysis due to reduced input size, and elimination of the overheads associated with decompression and recompression. Nevertheless, in the compression domain, original data become a sequence of binary bytes, or bits. Consequently, "understandability" based on some inherent patterns or correlations in the original data is lost. In addition to the difficulty of interpreting the compressed byte streams or bitstreams, conventional analysis algorithms that work well on regular data input will not function on data in their compression domain.

While compression domain data streams can be fully or partially decompressed to recover the original data, either in a lossy or lossless manner, the compression domain

process skips the decompression step and thus we need to have a good understanding of the compression process in order to be able to design analysis methods that take compressed data streams directly as inputs. To facilitate the discussion of the compression domain analysis methods surveyed, we provide brief introductions to the key techniques employed by the image and video compression standards, including the Joint Photographic Experts Group (JPEG) for image compression, and H.26x and several MPEG standards for video data compression.

The rest of the survey is organized as follows. Section 2 covers compression domain image data analysis. Section 3 discusses analysis methods for compressed video data. We summarize the survey in Section 4 and provide some thoughts on future research on compression domain processing and analysis.

2. Image

2.1. JPEG and DCT

Despite being introduced as early as the 1990s, the JPEG format [12] is still the most popular image format today. The acronym JPEG stands for the Joint Photographic Experts Group, which is a standards group under both the International Organization for Standardization and the International Electrotechnical Commission. The JPEG standard consists of methods and processes to compress digital images in a lossy manner. The baseline sequential process of JPEG compression is shown in Figure 1.

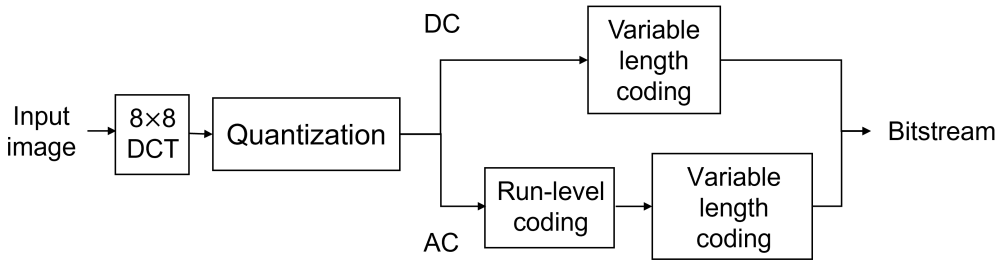

Figure 1. Flow chart for baseline JPEG for an image. An input image will go through multiple operations, including the discrete cosine transform (DCT), quantization, differential coding, and variable-length coding. The output is a bitstream with headers and markers containing side information for the decoder.

First, after a level shift, the image is divided into 8×8 minimum computing units (MCU). Each MCU is transformed by forwarding the DCT into an 8×8 DCT coefficients matrix according to Equations (1) and (2). The DC coefficient is at the top left, and the remaining 63 are AC coefficients. The 64-coefficient matrix is then quantized using a specified quantization table. Each value in the MCU is divided by the corresponding value in the quantization table and then rounded to the nearest integer. Next, the matrix is resized into a one-dimensional zig-zag sequence. The sequence of quantized DCT coefficients is then fed into an entropy encoder. All DC values are coded using a differential coding method. The 63 zigzag-ordered AC coefficients are coded using run-length coding. These result in a joint Huffman code of 8-bit values.

$$DC = \frac{1}{8} \sum_{m=0}^{7} \sum_{n=0}^{7} S_{mn}, \tag{1}$$

$$AC_{xy} = \frac{1}{4} \sum_{m=0}^{7} \sum_{n=0}^{7} S_{mn} cos \frac{(2m+1)x\pi}{16} cos \frac{(2n+1)y\pi}{16}. \tag{2}$$

During the process of JPEG compression, the DCT and quantization work jointly to reduce the data size, resulting in data compression. After the transform shown in the equations above, the input signal energy can be concentrated in just a few coefficients. The condensed information facilitates further processing to be performed on the DCT

coefficients, instead of on the pixels in the original image. Similar transform domain processing is employed by the more recent JPEG 2000 image compression standard, where wavelet transform is used instead of the DCT.

To facilitate a quick browse of the topics, we summarize in Table 1 the methods surveyed according to their applications.

Table 1. Summary of compression domain processing on image data.

Target	References	Detail
Image Resizing	[13–15]	Change image dimensions
Image Enhancement and Edge Detection	[16–24]	Change image properties to highlight certain region
Image Retrieval	[25–29]	Locate certain type of images from a large image database
Image Retargeting	[30,31]	Rearrange objects in differently sized images
Image Hiding	[32,33]	Conceal image in another image
Watermark Embedding	[34,35]	Add watermark on images
Image Classification	[36–39]	Separate images according to various attributes
Other Applications	[40,41]	Extract feature wavelet coefficients to detect objects, etc.

2.2. Image Resizing

Image resizing means converting an image of a given size to one of a different size. Generally, resizing includes zooming and shrinking. Zooming in the spatial domain is usually accomplished using interpolation algorithms, such as spline, bicubic, bilinear, etc. Shrinking can also be achieved using similar techniques, or by simply taking sample pixels on a fixed stride. There is also some work using neural networks to realize super resolution imaging, which may enable a new breakthrough in this direction.

In order to perform resizing operations directly in the compressed domain, DCT domain knowledge was first used in [13] by applying the convolution–multiplication property. The filtered coefficients were downsampled to the targeted size. Computing the new block produces the resized image. The downsampling filter was modified in [14], and the new half-reduced block is calculated by the following equation:

$$c = \sum_{i=1}^{4} h_i c_i g_i, \tag{3}$$

in which c_1 through c_4 are adjacent 8×8 blocks, h and g are downsampling filters given by

$$h_2 = g_1^t = g_3^t = h_1 = \begin{bmatrix} \mathbf{u}_{4\times 8} \\ \mathbf{o}_{4\times 8} \end{bmatrix}, h_4 = g_2^t = g_4^t = h_3 = \begin{bmatrix} \mathbf{o}_{4\times 8} \\ \mathbf{u}_{4\times 8} \end{bmatrix}, \tag{4}$$

$\mathbf{o}_{4\times 8}$ is a 4×8 zero matrix, and $\mathbf{u}_{4\times 8}$ is defined as

$$\mathbf{o}_{4\times 8} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}. \tag{5}$$

This method can preserve all the low-frequency DCT coefficients of the original image. Therefore, the resized image will have a better peak signal-to-noise ratio (PSNR) compared with other methods such as bilinear interpolation in the spatial domain. This result was further improved in [15] using sub-band DCT. For each 8×8 block B of the input image for halving, a 4-point inverse-DCT is applied with

$$A(k,l) = \frac{B(k,l)}{4cos\frac{\pi k}{2N} cos\frac{\pi l}{2N}}, \quad k,l = 0,1,2,3. \tag{6}$$

If we double the size of an image, then

$$A(k,l) = 4cos\frac{\pi k}{2N}cos\frac{\pi l}{2N}B(k,l), \quad k,l = 0,1,2,3\ldots 7. \tag{7}$$

The results showed that, although extra computation was involved, the overall performance was better than that of other existing methods.

2.3. Image Enhancement and Edge Detection

Image enhancement covers multiple image processing techniques intended to highlight certain information in the image, or diminish the impact of unwanted features such as noise or blur. Traditional image enhancement can be done in either the spatial domain or the frequency domain after applying a Fourier transform. Some common methods include contrast adjustment, histogram equalization, noise removal, smoothing and sharpening. On the other hand, edge detection locates the boundaries of objects in the image. The extracted features can be further used for image segmentation. Figure 2 shows how different operations would impact the output image.

Figure 2. Output for different image enhancement methods. The operations applied, from left to right, are: original image, histogram equalization, motion blur, deblur using Wiener filter, smoothing with Gaussian filter, image sharpening, edge detection with Sobel operator.

Both image enhancement and edge detection can be performed in the DCT domain, as shown in the following works. Shen and Sethi extracted features directly from DCT coefficients [16]. This is based on the fact that each DCT coefficient for any given 8×8 block is a linear combination of all the pixel values within the block. Therefore, the value of each AC coefficient reflects different grayscale values in a certain direction at a certain rate. Finally, the extracted information can be used for coarse edge detection in the original image, which can be 20 times faster than using the Sobel edge operator. Their work was extended in [17] to achieve faster convolution on DCT coefficient blocks, which further helped edge detection using a Laplacian-of-Gaussian operator. A comprehensive summary of these works can be found in [18]. The authors of [19] also proposed inner block transforms, which could realize regular geometric transformation by directly manipulating DCT domain data. The work was extended to compression domain video processing in [20].

DCT domain information was also used in [21] for edge enhancement of remote sensing image data. The algorithm consists of three parts: high pass filtering, adding back part or all of the gray levels to the original image, and contrast stretching. First, the 3×3 high pass filter kernel was decomposed using multiplication of matrices. The edge was enhanced by adding the gray levels and applying contrast stretching to the composite image. The same strategy was applied to edge enhancement of retina images in [22], in which 5×5 and 7×7 filter kernels were used for comparison.

Image segmentation can also be accomplished if the edges are located in the DCT domain [23], because variance in the block contains the edge information. For a given image block, the variance can be calculated using Equation (8). Later on, the whole image can be classified into three categories based on variance: homogeneous blocks, potentially high-frequency texture blocks, and edge blocks. Finally, region-growing techniques were applied to segment the original image.

$$\sigma^2 = \frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C^2(u,v) \quad (u,v) \neq (0,0). \tag{8}$$

Because of the sparsity of DCT coefficients after quantization, conducting image enhancement in the DCT domain can reduce storage requirements and computational expense [24]. The quantized coefficient blocks were separated into several bands along the anti-diagonal direction. Whether or not the output image is enhanced can be controlled by adjusting the parameters.

2.4. Image Retrieval

Image retrieval usually refers to the technique of retrieving images from a large image database. Common methods add extra information, such as captions or other keywords, so that retrieval can be completed using the corresponding annotations. One way of adding annotations is to automatically extract features from the image. The flowchart for a general image retrieval model is shown in Figure 3.

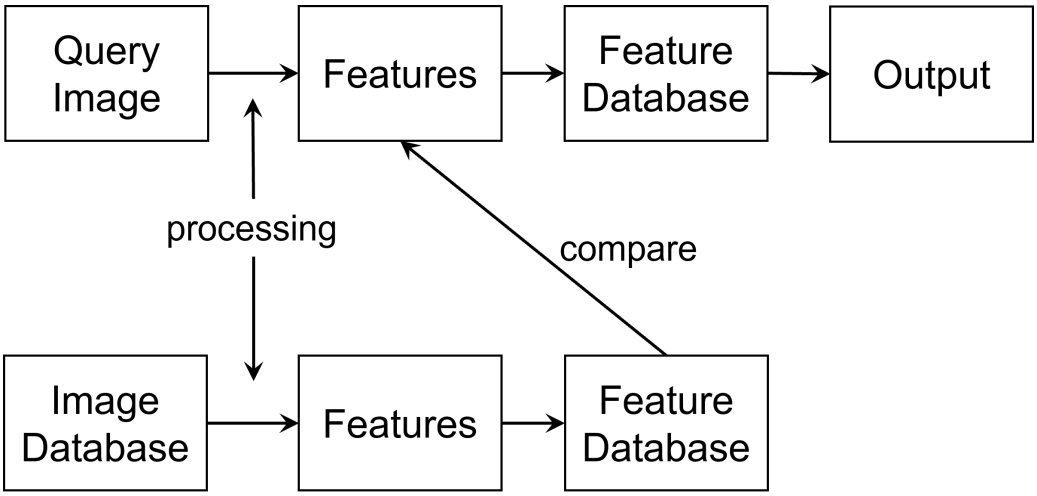

Figure 3. Flowchart for image retrieval.

For example, texture and color features extracted directly from the DCT domain were used in image retrieval in [25]. The absolute values of the AC components of the quantized DCT coefficients for each block were used as texture features. The DC components of the coefficients of the three channels Y, Cr and Cb were used to represent the color features.

Similarly, the DCT coefficients in YCrCb space were also used in [26]. The extracted feature vectors were clustered using the K-means method, with the centroids being the codewords for each image. The codebook, consisting of all codewords, is the histogram for each image in the database. The histogram can be used as a powerful tool to express the color distribution, which can further assist image retrieval.

Wavelet domain processing after a discrete wavelet transform (DWT) was also used in [27] to realize sketch-based retrieval; this was an extension of their previous work of using wavelet coefficients as compression domain index. On the other hand, conventional content-based image retrieval uses data from the pixel domain, which is time-consuming. Therefore, DCT coefficients were used in [28] to improve efficiency. The low-level features from dot-diffused block truncation coding bitmaps and high-level features from convolutional neural network (CNN) mode were combined to improve the overall accuracy for content-based image retrieval [29].

2.5. Image Retargeting

Unlike image resizing, which treats each pixel equally likely, image retargeting aims to preserve important regions in the image during resizing. A conventional retargeting flowchart is shown in Figure 4.

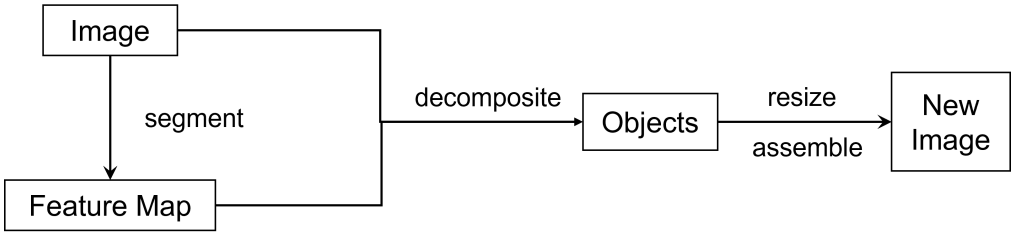

Figure 4. Flowchart for image retargeting.

This process can also be carried out in the compression domain. The intensity, color and texture features were extracted from DCT coefficients of the input image in [30]. The retargeting algorithm uses the saliency value of each DCT block via Hausdorff distance calculation and feature map fusion. Another multi-operator retargeting method uses indirect seam carving, similarity transformation, and direct seam carving of the DCT coefficients to perform resizing [31].

2.6. Image Hiding

Image hiding, sometimes called steganography, aims to hide an image (or information in other formats) inside another image. The hidden image should be able to be recovered at a relatively high quality. Traditional methods manipulate pixels to hide the extra information. Figure 5 shows a toy sample of the realization of image hiding by modifying the least significant bits. The underlying mechanic is that for a pixel of 8 bits, changing lower significance bits (the right four bits) will have much less impact on pixel value comparing with changing significant ones (the left four bits).

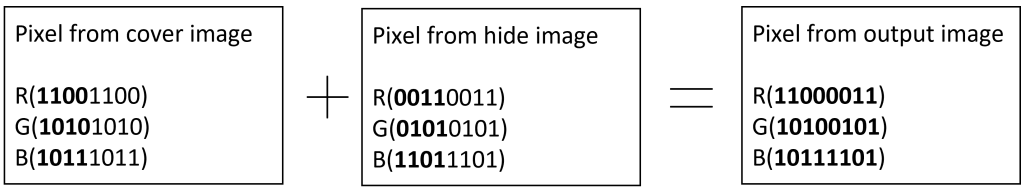

Figure 5. Toy sample for image hiding.

Unlike conventional data hiding, which directly embeds information bits into the bitstream, the method in [32] embeds Wiener filtering coefficients into the bitstream. Additionally, the filter could be further used to enhance the decoded images. A fairly comprehensive introduction to lossless information hiding in images can be found in [33]. The book covers information hiding techniques in three categories, i.e., the spatial, transform, and compression domains.

2.7. Watermark Embedding

Watermarking was invented to protect the copyright of targeted signals. Therefore, unlike image hiding, which tries to make hidden images imperceptible for human beings, watermark embedding aims at remaining robust against modification. The process of watermark embedding uses a watermarking key and watermarking algorithm to produce the watermarked digital image. Besides watermark embedding in the DCT domain, several other compression domain coefficients were also considered for watermark embedding in [34]. Transforms considered include the Karhunen–Loeve transform (KLT), Hadamard transform, wavelet transform and Slant transform. The results show that all of these methods have a larger hiding capacity when integrating watermarks into images. Patra et al. also applied Chinese Remainder Theorem in the DCT domain to ensure the robustness of watermarks [35]. The working process is shown in Figure 6.

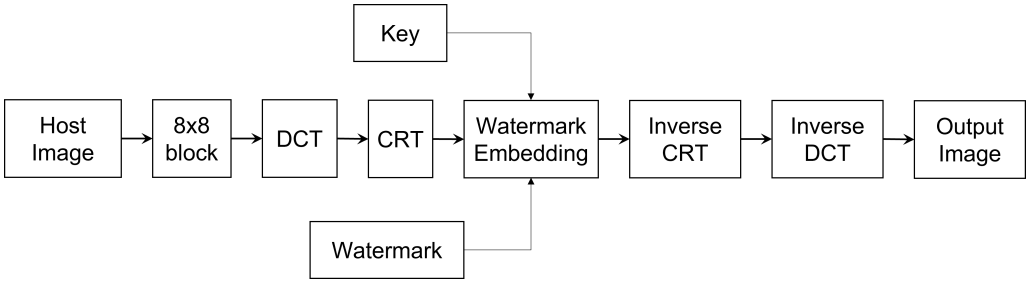

Figure 6. Overview of watermark embedding process.

2.8. Image Classification

Image classification aims to separate a group of images into several categories according to certain rules. Different features can be extracted from pixels in the image, serving as criteria for making the decision. Common methods that are currently widely used include K-nearest neighbor, support vector machine (SVM), and neural networks. A simplified flowchart is shown in Figure 7. For traditional methods, such as SVM, we need to generate features manually, which requires expertise in certain areas. However, with a neural network, features can be extracted automatically without domain knowledge.

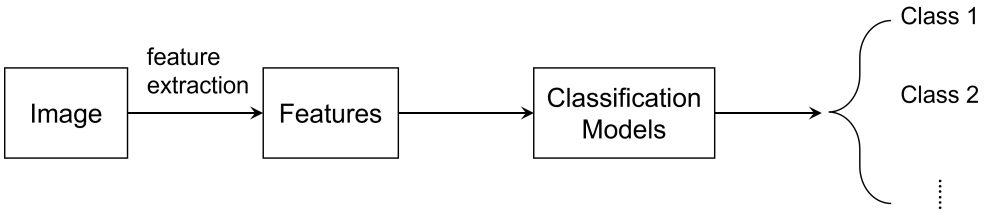

Figure 7. Simplified flowchart for image classification.

Although features from the spatial domain are more often used, features extracted from DCT coefficients can also be used for image classification. The work in [36] addresses screening out objectionable images to avoid under-age children seeing them (e.g., by identifying naked people in an image). Pixels in an image are classified as skin by the following calculation of the posterior probability of a pixel:

$$P(Skin|YCbCr, L) = \frac{P(YCbCr|Skin, L)}{P(YCbCr|Skin, L) + P(RGB|\overline{Skin}, L)} \geq \theta, \tag{9}$$

where $\theta \in [0, 1]$ is the threshold. The first term in the denominator is the prior probability of skin, and the second term is the non-skin pixels under average brightness L. Subsequently, the skin texture property is extracted from the lower frequency region in the DCT domain. Combined with other statistical features, the decision tree outputs the classification result.

After deep neural networks, such as convolutional neural networks (CNNs), gained popularity in image classification, the feasibility of using DCT coefficients was also examined. The two-dimensional DCT was applied directly to the input image in [37]. Then, a reduced size array was cropped from the coefficients matrix. The arrays were fed into a CNN for classification. The test showed that this method could speed up training time by a factor of 10, with a decrease in accuracy from 98% to 92% using the MNIST data set. This method provides users with a trade-off between accuracy and training time.

The application of multiple lossy compression operations to images is common in an image editing pipeline. Digital images can be easily used for the spread of false information, and thus their integrity needs to be questioned. The work in [42] addressed the forensic problem of classifying images based on the number of JPEG compressions they have gone through, by utilizing deep convolutional neural networks in the DCT domain. Handcrafted features, including the first 21 sub-bands (excluding the DC component) in the zig-zag order of the DCT coefficient matrix, were used for the histogram of the luminance channel.

For the chrominance channels, only the first three sub-bands were used. The combined feature vector was fed into the CNN model to estimate the number of compression operations. Experimental results showed that the algorithm could handle up to five rounds of compression.

In a recent study [38], DCT coefficients were also tested on MnasNet and Yolov5. The results showed a higher processing efficiency with a slight decrease in average precision. The original bitstream of JPEG was used in [39] for classification. The corresponding bitstream of each block was truncated or padded with zeros to maintain the same length. The 3D data were fed into CNN for classification.

The authors of this survey paper have also conducted research in this field [43], exploring the possibility of classifying images in their JPEG-compressed format. The study analyzed JPEG's underlying mechanisms and managed to separate malaria-infected red blood cells from normal cells based on information extracted from different stages of the JPEG compression process. The training data consisted of multiple combinations of DCT coefficients, DC values in decimal and binary forms, the "scan" segment in binary and decimal forms, and the bitstream at different lengths. The results showed that, using long short-term memory (LSTM), images could be successfully classified based on compression domain information with 80% accuracy. Moreover, accuracy of over 90% was attained simply by employing coded DC values, indicating that images from different classes could still be well distinguished in their JPEG-compressed format. Simulations demonstrated that the proposed method can significantly reduce the consumption of computational resources by shortening input data size and eliminating the image decompression step.

2.9. Other Applications

There are also plenty of other image processing areas that benefit from the compression domain approach. As an example, in ship detection using remote sensing techniques, spaceborne images are easily affected by clouds or ocean waves. Additionally, the processing efficiency can be low because of the high resolution of the images transmitted. The authors of [40] managed to combine wavelet coefficients extracted from the JPEG2000 compression domain, deep neural network and extreme learning machine to solve these issues. Instead of using DCT coefficients in JPEG, DWT coefficients are used to divide the land and sea, providing ship candidates. Next, the autoencoders are trained to extract meaningful features from the high and low wavelet coefficients. Finally, the sequential extreme learning machine is trained to make the decision.

Another example is face recognition using features extracted from DCT coefficients of normalized images in [41]. It was shown that, compared with Karhunen–Loeve transform, DCT could have desirable pattern recognition capabilities. Additionally, the light computational complexity of DCT makes it suitable for face recognition. The classification was performed using a simple Euclidean distance measure of the features vector containing reduced-size DCT coefficients.

Generating visual data from text could be a difficult task in the past. With the help of Generative Adversarial Networks (GANs), the model proposed in [44] can be trained directly on DCT coefficients and achieve good performance. The input descriptive text data was first represented as numerical vectors, using text embedding. After noise is added, the image tensor can be obtained by applying several convolutions and normalizations. For this DCT-based generator network, the tensor was divided into three channels, corresponding to the Y, Cb and Cr channels. The locally connected layer would produce amplitudes of DCT for each channel, and then quantization was applied to generate the DCT-compressed images.

On the contrary, another group of researchers managed to learn semantics from DCT-based frequency domain representation in [45]. They showed that it is possible to preserve meaningful semantic information, even when high-frequency components in DCT coefficients are dropped. The frequency domain compression was conducted on the DCT coefficients matrix, which only keeps the DC component, most of the low-frequency and

some middle-frequency information. After augmentation using positional and frequency embedding, the matrix was fed into blocks for classification with linear projection.

2.10. Summary

Although we have categorized algorithms according to their applications, different methods may share more in common than is initially apparent. Firstly, most existing image processing techniques in the compression domain use DCT coefficients as input. This is because DCT is the core component of JPEG, which continues to be the most popular image compression format. However, we should also note that, for applications aiming to save computational resources with compressed images, DCT coefficients are not the best option because DCT is conducted at the end of decompression, which means that most of the resource-consuming steps have been done. Secondly, different application scenarios may share similar underlying image processing algorithms. Image retargeting inevitably uses image resizing to generate suitable sizes for objects in the output image. Image retrieval and classification both take features extracted from images and use the new tags to separate one image from another. Finally, we have to admit that some of the image processing methods discussed are outdated. Therefore, more in-depth research should be carried out on the compressed bitstream, which requires no decompression, and more advanced image compression formats such as WebP.

3. Video

3.1. Video Compression and IPB Frames

In a video stream, each frame can be also be viewed as an image. Video compression relies on exploiting the correlations between adjacent frames using motion estimation and motion compensation. There are three major types of frame: I, P and B frames. I stands for intra-coded pictures, which can be decoded without other frames. P denotes frames that are predicted using the previous frame in video compression. Therefore, decoding a P frame requires the information of a decompressed previous frame. B stands for bidirectional predicted frames, decoding of which requires both the preceding and following frames. P and B frames are also called inter frames. A simple illustration of the IPB frames in a video sequence is shown in Figure 8.

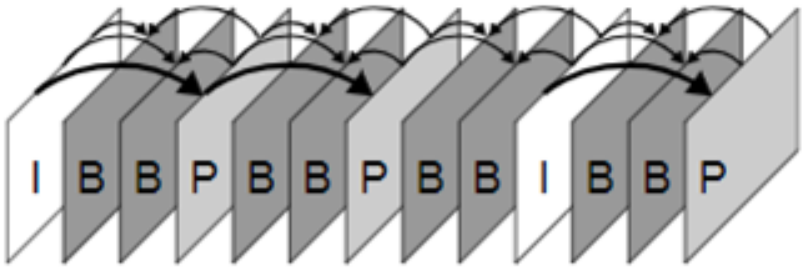

Figure 8. Intra (I) and inter (B and P) frames in a video sequence.

The Moving Picture Experts Group (MPEG) is another working group similar to JPEG, with a focus on the efficient compression of video data. Later, MPEG was used as the name of a video format after compression. There are several well known MPEG video formats such as MPEG-1, MPEG-2 and MPEG-4. H.264, also known as Advanced Video Coding (AVC) or MPEG-4 Part 10, is currently the most widely used video compression standard. H.265, also known as High-Efficiency Video Coding (HEVC) or MPEG-H Part 2, is a newer video compression standard, designed as an upgrade of H.264. However, H.265 requires much more processing power and advanced hardware, thereby limiting its popularity. The flowchart of MPEG-1 encoding is shown in Figure 9.

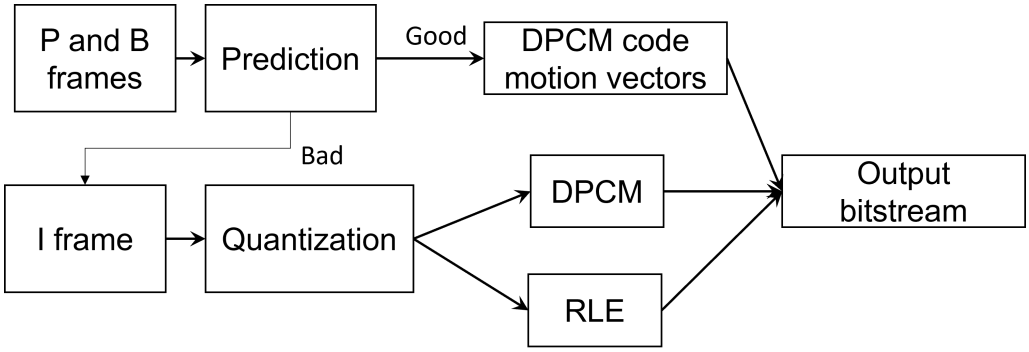

Figure 9. Flowchart of MPEG-1 encoding.

Some drawbacks of MPEG-1 were overcame in MPEG-2. For example, instead of only supporting progressive scanning, MPEG-2 could also handle interlace scanning well, and had the capability to compress high-resolution videos. After MPEG-3 was abandoned, MPEG-4 was optimized to achieve higher compression efficiency and video quality compared with its predecessor. One major improvement is that MPEG-4 uses 16 × 16 DCT, instead of the traditional 8 × 8 in MPEG-2 or JPEG. This allows for a larger compression ratio. The computing unit was further improved in H.265, which started to use coding tree units (CTUs). Ranging from 4 × 4 to 64 × 64, the different sizes of macroblock allow the algorithm to process data more efficiently. In H.266, the size was increased to 128 × 128 to provide the codec with additional flexibility.

Most processing and analysis techniques in the video compression domain rely on motion vectors, which are two-dimensional vectors used for the prediction of inter video frames. Motion vectors are the result of motion estimation conducted by a video encoder. The decoder performs motion compensation by using motion vectors as offsets to locate similar blocks from a reference frame in order to predict the current frame.

Table 2 summarizes various applications of compression domain processing and analysis methods surveyed.

Table 2. Summary of compression domain processing and analysis on video data.

Target	References	Details
Video Compositing	[46–51]	Combine multiple video stream into one
Video Retrieval	[52–55]	Locate certain type of videos from a large video database
Watermark Embedding	[56–59]	Add watermark into video
Video Transcoding	[60–63]	Convert video from one format to another
Object Detection	[64–70]	Locate certain object
Segmentation	[71–74]	Separate certain object from frames
Video Steganalysis	[75,76]	Hide information in video stream
Salient Motion Detection	[77–80]	Detect saliency in video
Video Resizing	[81–83]	Change dimension of video
Video Summarization	[84–86]	Add tags that can be representative of the video
Other Applications	[87–97]	Detect double compression, etc.

3.2. Video Compositing

Videos are usually transmitted in a compressed form over networks. There are situations in which a multi-user video network server wants to combine multiple compressed video sources into a single compressed output stream. This is where video compositing can be implemented. The ability to directly compose video streams in the DCT domain will significantly speed up processing. In [46], the authors converted all MC-DCT compressed video into the DCT domain and performed compositing in the DCT domain. The model used is shown in Figure 10.

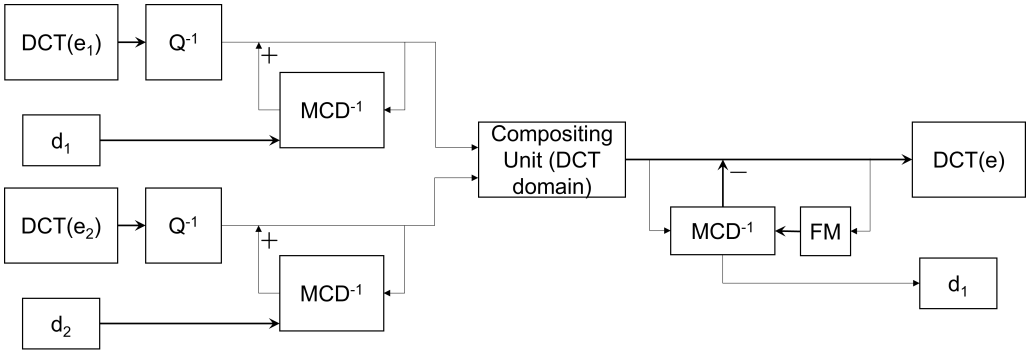

Figure 10. Compositing two DCT-based video sequences in DCT domain.

The algorithm was improved in [47] with a fast algorithm that converts motion-compensation compressed video into a sequence of DCT-domain blocks. These blocks correspond to the spatial domain blocks of the current frame, eliminating the need to use other reference frames for prediction. As a result, the inter-frame element of the compression–decompression pipeline is removed. This method enables not only video compositing in the DCT compressed domain, but also allows for several other operations, e.g., scaling, overlapping, translation, filtering, etc. Similar video processing systems and related tools can be found in [48,49]. An algorithm designed for MPEG1 video compositing based on the same idea was proposed in [50].

In [51], the authors tackled the challenge of handling motion-JPEG video data in the compressed domain. It was shown that, where pixels in the output image are linear combinations of those from the input, several operations can be performed in the compression domain. The verified operations include convolution, scaling, rotation, translation, morphing, de-interlacing, image composition, and transcoding. It was observed that image compression, decompression and other operations can be treated as tensors. All of these tensors can be combined to construct a single linear operator that can be applied to compressed video images. An approximation technique called condensation was introduced in [51]. The idea is to approximate compression domain operators so that they can be efficiently computed. Condensation modifies an operator to a sparse operator, so that the effect of the sparse operator result will be nearly identical to that of the original operator. Condensation would increase processing speed at the cost of slight degradation in the quality of processed images.

3.3. Video Archiving, Indexing and Retrieval

Similar to image retrieval, video archiving, indexing and retrieval also require adding tags to each piece in a large video database. Utilizing features that are automatically created from images can significantly minimize human effort. The basic flowchart is also identical to that of image retrieval. The only difference is that instead of extracting features from images, in this case, the algorithm extracts features from key frames.

Reviews of outdated image and video indexing techniques can be found in [54,55], which can provide readers with some background information about these topics. Editing and parsing digital video directly in the compression domain has many advantages in terms of storage efficiency, speed and video quality [48,49]. Compression domain parsing of video relies on shot change detection and motion detection using data in the compression domain. To detect shot changes, [52] used features derived from the available DCT coefficients, macroblocks, and motion vector information. More features were extracted from DCT coefficients and motion vectors in MPEG video for better performance [53].

3.4. Caption and Watermark Embedding

Rather than adding a watermark to one image, watermark embedding in video requires the operation to be performed on multiple frames. Adding descriptive text to video frames is a useful feature of a video editor. Traditional watermark embedding requires the

raw video sequence to be recovered first, the watermark added, and then encoded again, as shown in Figure 11.

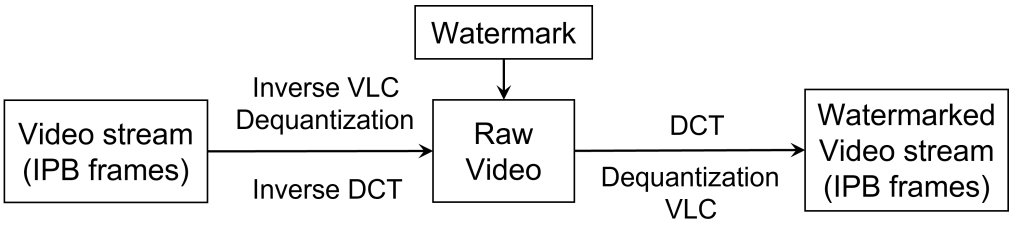

Figure 11. Watermark embedding in spatial domain.

A stochastic approximation model was used in [56] to support caption processing in the MC-DCT domain. The strength of the added text is determined by the mean and variance of the input image block. Then, the DCT coefficients of the added text can be inserted into the video in the DCT domain. The mean and variance of the input blocks can be derived using Equation (10), where Y_{DC} and Y_{AC} are the DC and AC coefficients of the input image block Y.

$$\begin{aligned} \alpha &= Y_{DC}/8, \\ \beta^2 = Var(y) &= \frac{\sum_{l=0}^{63} Y_l^2}{64} - \frac{Y_{DC}^2}{64} = \frac{\sum_{l=1}^{63} Y_l^2}{64} = \frac{\sum Y_{AC}^2}{64} \end{aligned} \tag{10}$$

A similar method was adopted in [57] to add captions on a sequence of video frames. The DC value of the input block was used as the approximated value for all pixels. This method can avoid having watermark mask values that exceed the maximum value allowed in the MC-DCT domain.

Video watermark embedding/extracting in the H.264 compression domain was proposed in [58], in which video bitstream was utilized to avoid extensive compression and decompression. The activities of each block were determined by the number of nonzero (NNZ) entries of quantized AC residuals. Therefore, desynchronization issues can be eliminated as long as the embedding and extracting are restricted to areas of high spatial activity. It was shown that these macroblocks are robust against intraprediction mode changes.

As a relatively new video compression standard, HEVC could also benefit from watermarking in the compression domain, as shown in [59]. The embedding was only performed in P frames to minimize the degradation of output video quality. Security and robustness can be guaranteed by random selection and the spatiotemporal characteristics of the compressed video. As in [58], the blocks with higher NNZ may be selected for watermark embedding.

3.5. Video Transcoding

Video transcoding is the conversion of video data from one format to another. This function is vital when a target device does not support the current video format, or has limited storage capacity that requires a reduced file size. Transcoding is also used to convert incompatible or obsolete video data to a newer, more widely supported video format. The general procedure is similar to that shown in Figure 12.

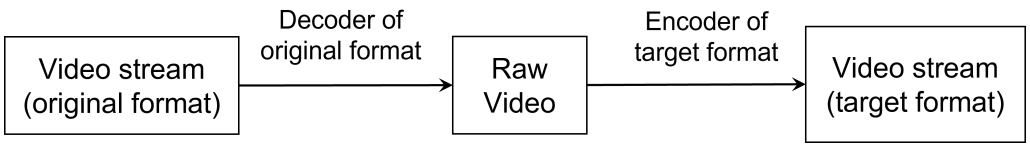

Figure 12. Conventional video transcoding flowchart.

Unlike previous work that required full decoding, the approach proposed in [60] used a Huffman decoded bitstream to transcode MPEG-1 video to motion-JPEG. This method provided faster processing at the cost of lower picture quality. Another two transcoding schemes were proposed in [61]. The first one simply implemented both MC (motion compensation) loops in the DCT domain. In the second scheme, image downsampling was used to build a hybrid DCT-spatial domain architecture. Fast refinement for non-integral motion vectors was introduced in [62] to provide better quality and lower complexity for a spatial-downscaling video transcoder in the DCT domain.

A special case of transcoding is reverse play, which can produce a video stream in the reverse order after processing by the transcoder [63]. Unlike the spatial method that needs decoding, reordering and re-encoding, transcoding in the compression domain uses a forward motion vector in the regular video bitstream to create the reverse motion vector. Therefore, the computational complexity can be greatly reduced.

3.6. Object Detection

Information extracted from DCT coefficients and motion vectors in the video compression domain can also be used for object detection in video. The general processing flow is illustrated in Figure 13.

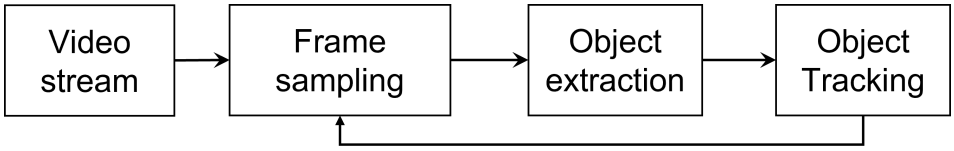

Figure 13. Overview of the object tracking method.

This idea gave rise to the work in [64], which extracted metadata from video sequences in the MPEG-2 compressed domain. Metadata include camera motion, regions of interest to be tracked, and scene cuts to be detected. For object detection, the DCT coefficients of I frames and intracoded macroblocks of P frames were used directly from the MPEG-2 compressed data. For non-intracoded ones, the DCT block calculation can be simplified by approximating the four adjacent blocks, because only information concerning whether a macroblock is sufficiently textured really matters.

The motion vectors in a compressed HEVC bitstream are good enough to indicate the approximate location of the target object [65]. Then, a more accurate location can refine the bounding box in the decoded frame.

A convolutional neural network (CNN) was also used, along with motion vectors, for real-time object tracking in [66]. For independent frames, such as I frames, CNN was applied for accurate detection. Meanwhile, for dependent frames, such as P frames, interpolation was applied using the detection results of I frames. This method was extended in [67] to predict the velocities of objects by considering the scale variation of bounding boxes. The features used in [68] included three types of data: partitioning depths, prediction modes, and residuals. All of these were extracted from I frames. The pixel-level resolution of residuals guarantees the pixel-level object localization.

Using features derived from motion vectors of HEVC compressed video sequences, a moving object detection method was proposed in [69]. This method was developed upon the conditional random field [70], which was updated for every frame. Like hidden Markov models, the conditional random field is a probabilistic model for segmenting and labeling sequence data.

Face detection directly from the HEVC bitstream was addressed in [73]. The feature map fed into a CNN consisted of three feature channels extracted from entropy decoding.

Video action recognition can be viewed as an extension of object detection. Instead of using decoded RGB frames, the model proposed in [98] only operates on I frames, motion vectors, residuals and audio waveforms. The efficiency of the model is ensured by the features' ease of access.

3.7. Segmentation

Detecting the constituent parts of video frames enables us to partition them into multiple segments or objects, which is a critical aspect of various practical applications such as enhancing visual effects in movies, understanding scenes in autonomous driving, and creating virtual backgrounds in video conferencing. However, conventional video segmentation approaches are limited to the spatial domain, in which no motion information is available. To estimate motion, methods such as block matching, phase correlation, and gradient-based approaches are commonly employed on pairs of consecutive frames. An alternative approach to improve video segmentation is to leverage the advantages of compressed video, as motion cues are already present in the P frames. In practical situations, segmentation is often accompanied by object detection.

Along the same lines, compression domain segmentation methods were proposed in [71] to achieve results comparable to those of more computationally expensive segmentation methods on raw data.

A compression domain-based propagation method using a deep CNN was employed in [72] to achieve real-time video segmentation. This method was proposed to expedite inference speed for semi-supervised video object segmentation tasks. The deep CNN was used to extract features of I frames, and information flow propagation was employed to generate features of P frames.

The study in [74], focusing on semantic video segmentation, proposed a new method with three modules to improve accuracy while reducing noise caused by motion vectors. These modules included a feature warping module, a residual-guided correction module for refinement, and a residual-guided frame selection module. The results showed that the proposed modules successfully achieved acceptable accuracy degradation while improving the overall performance.

3.8. Video Steganalysis

Video steganography is a branch of data hiding that embeds information into cover video contents in such a manner that the presence of the information is not evident to human inspection. The video steganalysis method proposed in [75] aimed to detect information hidden in videos. The pair of conditional and joint distributions of the adjacent difference in the DCT and DWT (discrete wavelet transform) domains was extracted to be the feature used for classification. A comprehensive survey of both compression domain and raw video stream steganography techniques can be found in [76]. Another survey paper, [99], also analyzed video steganography over uncompressed and compressed domains and may offer a wider view of this specific problem.

3.9. Salient Motion Detection

The analysis of surveillance video is a growing area of research in computer vision. Motion saliency denotes the conspicuous state of an object in a video. Detecting motion saliency relies on motion detection. One way to detect salient motion is shown in Figure 14.

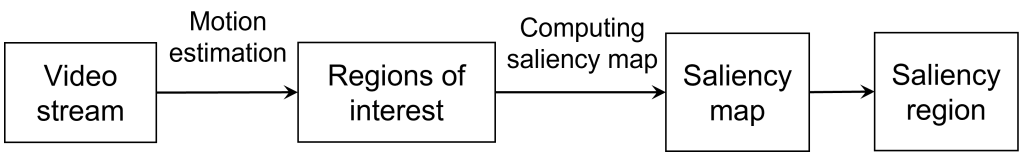

Figure 14. Flowchart for salient motion detection.

The DCT coefficients of luminance and chroma components were used in [77] to calculate the spatial saliency of a given frame. The motion vectors were also used to refine the results. This two-step method could measure saliency in video frames without camera motion. Similar work was also done in [78] with a different selection of features.

A CNN was involved in [79] to obtain the saliency area. Combined with motion estimation results of each block during HEVC compression, the algorithm completed the

saliency map of the input video. A 3D deep CNN was applied in [80] to extract features from the motion vectors of the macroblocks in the P frames for video classification. The extra dimension was time.

3.10. Video Resizing

A compression domain approach was also used in [81] to realize down-conversion from the Common Intermediate Format (CIF) to the Quarter Common Intermediate Format (QCIF) bitstream. Compared with conventional methods in the pixel domain, compression domain approaches could eliminate three computationally expensive blocks, i.e., DCT, IDCT and motion estimation. A similar strategy was also applied in [82], in which some high-order AC coefficients were discarded, providing the flexibility to choose between computational cost and video quality. Unlike the above methods, the method proposed in [83] used three macroblocks instead of four while providing comparable quality.

3.11. Video Summarization and Abstraction

It would be desirable to be able to understand the content of a video without having to watch the entire video. Video summarization tends to create a short synopsis that summarizes the important parts of the stream. Video summarization and abstraction allows for quick indexing, searching, browsing and evaluation of the input data. A simplified model for video summarization can be found in Figure 15.

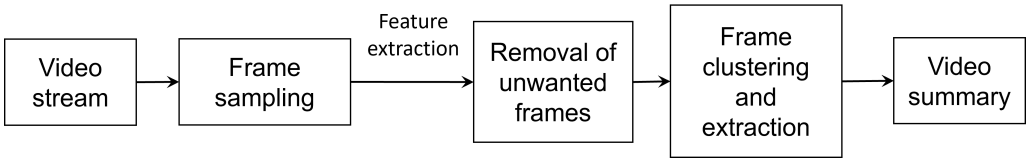

Figure 15. Flowchart for video summarization.

The work in [84] addressed the progressive generation of a video summary in the compression domain. The proposed method relies on exploiting visual features extracted from the video stream and using a simple and fast algorithm to summarize the video content. Specifically, the DC value of each DCT block was extracted to build DC images, which serve as the criteria for selecting representative frames. This method was further developed in [85], which used I frames of HEVC coded video for video abstraction in the compression domain. Clustering was also applied to extract the key frames. For more work on video summarization techniques in the compression domain, see [86].

3.12. Other Applications

Motion estimation in compressed bitstreams of fingerprint videos helped in the detection of dynamic behavior in [87], where the proposed method could reliably detect and characterize distortion. In [88], a representative frame was selected, and then the corresponding DCT coefficients were used to detect scene change and to select sub-regions for subsequent processing and analysing. Ahmed [89] developed a programmable video co-processor that can handle DCT-domain operations including resolution conversion, frame rate changing, quality and rate control, filtering, video compositing and video cut detection.

The video encoder can also be improved in the compression domain [90]. The low-resolution coarse-step motion estimation operations were performed in the DCT domain. High-level motion activity could also be determined by computing the block matching in the reference frame, at the cost of slightly increased computational complexity. Another low-bit-rate video coding algorithm was proposed in [91]. DCT-decimation was used in the encoder, while the corresponding DCT-interpolation was used in the decoder. Embedded zerotree [92] and an adaptive arithmetic coder were also used in the encoder to improve the quality of the decoded video.

Unlike the previous studies, which focused on lossy compression formats, [93] presents a technique for translating a specific class of computations to operate directly

on losslessly-compressed data using the sliding window Lempel–Ziv algorithm. The applications of this technique include video editing and compositing directly on losslessly compressed video.

Video encryption can also be performed in the DCT domain. The method proposed in [94] extracted the most significant bits for video reconstruction in H.264 streaming and concatenated them into a sub-bitstream. The encrypted bitstream was then rearranged into the original positions.

Compression-domain highway vehicle counting using spatial and temporal regression was proposed in [95], in which low-level features were used to capture crucial information that is useful for counting vehicles. These features can be computed from the motion vectors and block partition modes. The features includes the size, shape, motion, and texture information of traffic scenes. These features can then be combined to train a hierarchical classification-based regression model to count vehicles in a video frame.

To detect double HEVC compression, an attention-based two-stream residual network was proposed in [96] to build a hybrid neural network architecture together with LSTM. This method could efficiently learn spatio-temporal representations of relocated I frames, and was able to achieve high robustness against recompression.

Super-resolution aims to achieve high-resolution data streams from their low-resolution observations. The spatial and temporal coding priors extracted from the compression domain were used in [97] as the input for a deep neural network. The proposed scheme sought to achieve video super-resolution and the suppression of compression artifacts in an end-to-end manner.

3.13. Summary

Compared with 2D images, frames in video carry much more information. However, we can still find correlations between these two data formats. The original MPEG-1 format borrowed almost all its techniques from JPEG. Therefore, most of the algorithms that used to work in the compression domain for images can be used for video as well (possibly with minor modifications). Even for an enlarged macroblock of 16×16, the DCT coefficients still carry information that represent original frames. It is also important to note that the motion vectors are used quite often because they can be easily accessed from the compressed bitstream without verbose decoding procedures. Although we have classified different methods in separate sections, they are actually closely related. For object detection, segmentation and salient motion detection, certain features or regions need to be extracted. Video archiving, indexing, retrieval and summarization require the gathering of important frames that can be representative. Therefore, understanding the mechanism of any one algorithm may help develop more efficient new methods.

4. Discussion

In this survey, we presented a comprehensive review of existing work in the literature concerning compression domain techniques for video and image data. Processing and analysis of data in their compressed forms can greatly reduce the size of input data and eliminate the need for decompression and recompression, thereby achieving significant savings in memory and computation time. We can see that different methods make use of different entities in the compression domain, including DCT coefficients, wavelet transform coefficients, motion vectors, etc.

Although numerous papers cited here used DCT coefficients as their input, this is not the optimal solution for compression domain analysis, as we mentioned earlier. The reason is that in both image and video decompression, inverse DCT occurs last, which means most heavy jobs have been done. Therefore, the primary goal for using compression domain data to save computational resources and time—is not fully attained. The compressed bitstream should take the priority, because extracting information from a bitstream requires no decompression.

Additionally, with the great success achieved through the application of deep neural networks in image and video processing, some of the older compression domain techniques have been revisited and revised according to the new framework of deep learning approaches. However, the underlying mechanisms of image and video compression remain unchanged. Therefore, further research into how to properly use features from compressed bitstreams in the deep learning framework is warranted.

On the other hand, there has been little work on compression domain processing based on compression formats other than JPEG, MPEG, and H.26x. Examples include lossless image compression formats based on Portable Network Graphic (PNG). New challenges are on the horizon with the appearance of the H.266 video format. This so-called Versatile Video Coding is an up-and-coming video compression standard. Early tests suggest that H.266 achieves 40–50% better compression than HEVC codecs. We anticipate a lot of new compression-domain work with this new video format.

5. Conclusions

Previous researchers have made significant contributions to the field of image and video processing with compression domain data. However, as we are now creating larger amounts of data with more advanced compression techniques, more in-depth research should be conducted on the real compression domain data, bitstreams. The fundamental theorem for state-of-the-art algorithms should also be carefully explored. A combination of previous research with a modern neural network model may result in a surprising discovery.

In closing, we point interested readers to more references. Ref. [100] is an introductory-level book focused on the DCT (used in JPEG, MPEG and H.264) and DWT (used in JPEG2000). It also covers various image and video processing operations performed in the compression domain, including filtering, enhancement, color restoration, resizing, transcoding, watermarking, indexing, face detection, steganography, etc. Ref. [101] is a more recent survey paper focusing on video analysis techniques in the compression domain. The techniques used for document image analysis in the compression domain are discussed in [102].

Author Contributions: Conceptualization, Y.D. and W.D.P.; methodology, Y.D. and W.D.P.; validation, Y.D. and W.D.P.; formal analysis, Y.D. and W.D.P.; investigation, Y.D. and W.D.P.; resources, Y.D. and W.D.P.; data curation, Y.D. and W.D.P.; writing—original draft preparation, Y.D. and W.D.P.; writing—review and editing, Y.D. and W.D.P.; visualization, Y.D. and W.D.P.; supervision, W.D.P.; project administration, W.D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

JPEG	Joint Photographic Experts Group
DCT	Discrete cosine transform
MCU	minimum coded unit
PSNR	peak signal-to-noise ratio
DWT	discrete wavelet transform
SVM	support vector machine
CNN	Convolutional neural network
LSTM	Long short-term memory

MPEG	Moving Picture Experts Group
HEVC	High Efficiency Video Coding
MV	motion vector
MC	motion compensation
NNZ	number of nonzero
PNG	Portable Network Graphic
CTU	coding tree unit

References

1. Paula Dootson. 3.2 Billion Images and 720,000 Hours of Video Are Shared Online Daily. Can You Sort Real from Fake? Available online: https://www.qut.edu.au/study/business/insights/3.2-billion-images-and-720000-hours-of-video-are-shared-online-daily.-can-you-sort-real-from-fake (accessed on 14 March 2023).
2. Antonio, R.; Faria, S.; Tavora, L.M.; Navarro, A.; Assuncao, P. Learning-based compression of visual objects for smart surveillance. In Proceedings of the 2022 Eleventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Salzburg, Austria, 19–22 April 2022; IEEE: New York, NY, USA,2022; pp. 1–6.
3. Bhardwaj, V.; Rasamsetti, Y.; Valsan, V. Traffic Control System for Smart City Using Image Processing. In *AI and IoT for Smart City Applications*; IEEE: New York, NY, USA, 2022; pp. 83–99.
4. Mavrogiorgou, A.; Kiourtis, A.; Kyriazis, D. Iot devices recognition through object detection and classification techniques. In Proceedings of the 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), London, UK, 30–31 July 2019; IEEE: New York, NY, USA, 2019; pp. 12–20.
5. Anand, A.; Singh, A.K.; Lv, Z.; Bhatnagar, G. Compression-then-encryption-based secure watermarking technique for smart healthcare system. *IEEE Multimed.* **2020**, *27*, 133–143. [CrossRef]
6. Ammah, P.N.T.; Owusu, E. Robust medical image compression based on wavelet transform and vector quantization. *Inform. Med. Unlocked* **2019**, *15*, 100183. [CrossRef]
7. Abdellatif, A.A.; Emam, A.; Chiasserini, C.F.; Mohamed, A.; Jaoua, A.; Ward, R. Edge-based compression and classification for smart healthcare systems: Concept, implementation and evaluation. *Expert Syst. Appl.* **2019**, *117*, 1–14. [CrossRef]
8. Pareek, P.K.; Sridhar, C.; Kalidoss, R.; Aslam, M.; Maheshwari, M.; Shukla, P.K.; Nuagah, S.J. IntOPMICM: Intelligent medical image size reduction model. *J. Healthc. Eng.* **2022**, *2022*, 5171016. [CrossRef]
9. Dimililer, K. DCT-based medical image compression using machine learning. *Signal Image Video Process.* **2022**, *16*, 55–62. [CrossRef]
10. Golini, M. *Real-Time and High-Quality Video Compression for Telesurgery*; Politecnico di Milano: Milan, Italy, 2022.
11. Sikka, R. Agricultural Image Analysis on Wavelet Transform. In *Proceedings of the International Conference on Intelligent Emerging Methods of Artificial Intelligence & Cloud Computing: Proceedings of IEMAICLOUD 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 122–127.
12. Wallace, G.K. The JPEG still picture compression standard. *Commun. ACM* **1991**, *34*, 30–44. [CrossRef]
13. Martucci, S.A. Image resizing in the discrete cosine transform domain. In *International Conference on Image Processing*; IEEE: New York, NY, USA, 1995; Volume 2, pp. 244–247.
14. Dugad, R.; Ahuja, N. A fast scheme for image size change in the compressed domain. *IEEE Trans. Circuits Syst. Video Technol.* **2001**, *11*, 461–474. [CrossRef]
15. Mukherjee, J.; Mitra, S.K. Image resizing in the compressed domain using subband DCT. *IEEE Trans. Circuits Syst. Video Technol.* **2002**, *12*, 620–627. [CrossRef]
16. Shen, B.; Sethi, I.K. Direct feature extraction from compressed images. In Proceedings of the Storage and retrieval for still image and video databases IV, San Jose, CA, USA, 28 January–2 February 1996; SPIE: Washington, DC, USA, 1996; pp. 404–414.
17. Shen, B.; Sethi, I.K. Convolution-based edge detection for image/video in block DCT domain. *J. Vis. Commun. Image Represent.* **1996**, *7*, 411–423. [CrossRef]
18. Shen, B. *Compressed Domain Processing: Algorithms and Applications*; Wayne State University ProQuest Dissertations Publishing: Detroit, MI, USA, 1997.
19. Shen, B.; Sethi, I.K. Block-based manipulations on transform-compressed images and videos. *Multimed. Syst.* **1998**, *6*, 113–124. [CrossRef]
20. Wee, S.; Shen, B.; Apostolopoulos, J. Compressed-Domain Video Processing. In *Hewlett-Packard, Tech. Rep. HPL-2002-282*; 2002. Available online: https://www.hpl.hp.com/techreports/2002/HPL-2002-282.pdf (accessed on 14 March 2023).
21. Chen, B.; Latifi, S.; Kanai, J. Edge enhancement of remote sensing image data in the DCT domain. *Image Vis. Comput.* **1999**, *17*, 913–921. [CrossRef]
22. Javed, M.; Nagabhushan, P.; Chaudhuri, B.B.; Singh, S.K. Edge based enhancement of retinal images using an efficient JPEG-compressed domain technique. *J. Intell. Fuzzy Syst.* **2019**, *36*, 541–556. [CrossRef]
23. Jiang, J. Image segmentation in compressed domain. *J. Electron. Imaging* **2003**, *12*, 390. [CrossRef]
24. Tang, J.; Peli, E.; Acton, S. Image enhancement using a contrast measure in the compressed domain. *IEEE Signal Process. Lett.* **2003**, *10*, 289–292. [CrossRef]
25. Jain, A.K.; Zhong, Y.; Jain, A.K. Object localization using color, texture and shape. *Pattern Recognit.* **2000**, *33*, 671–684.

26. Jamil, A.; Majid, M.; Anwar, S.M. An Optimal Codebook for Content-Based Image Retrieval in JPEG Compressed Domain. *Arab. J. Sci. Eng.* **2019**, *44*, 9755–9767. [CrossRef]
27. Pimentel Filho, C.A.F.; Bustos, B.; Araújo, A.d.A.; Guimarães, S.J.F. Combining pixel domain and compressed domain index for sketch based image retrieval. *Multimed. Tools Appl.* **2017**, *76*, 22019–22042. [CrossRef]
28. Temburwar, S.; Rajesh, B.; Javed, M. Deep Learning-Based Image Retrieval in the JPEG Compressed Domain. In *Advanced Machine Intelligence and Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 351–363.
29. Liu, P.; Guo, J.M.; Wu, C.Y.; Cai, D. Fusion of deep learning and compressed domain features for content-based image retrieval. *IEEE Trans. Image Process.* **2017**, *26*, 5706–5717. [CrossRef]
30. Fang, Y.; Chen, Z.; Lin, W.; Lin, C.W. Saliency detection in the compressed domain for adaptive image retargeting. *IEEE Trans. Image Process.* **2012**, *21*, 3888–3901. [CrossRef]
31. Tang, Z.; Yao, J.; Zhang, Q. Multi-operator image retargeting in compressed domain by preserving aspect ratio of important contents. *Multimed. Tools Appl.* **2022**, *81*, 1501–1522. [CrossRef]
32. Jung, S.W. Adaptive post-filtering of JPEG compressed images considering compressed domain lossless data hiding. *Inf. Sci.* **2014**, *281*, 355–364. [CrossRef]
33. Lu, Z.M.; Guo, S.Z. *Lossless Information Hiding in Images*; Zhejiang University Press: Hangzhou, China, 2016.
34. Fei, C.; Kundur, D.; Kwong, R. The choice of watermark domain in the presence of compression. In Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 2–4 April 2001; IEEE: New York, NY, USA, 2001; pp. 79–84.
35. Patra, J.C.; Phua, J.E.; Bornand, C. A novel DCT domain CRT-based watermarking scheme for image authentication surviving JPEG compression. *Digit. Signal Process. A Rev. J.* **2010**, *20*, 1597–1611. [CrossRef]
36. Ye, Q.; Gao, W.; Zeng, W.; Zhang, T.; Wang, W.; Liu, Y. Objectionable image recognition system in compression domain. *Lect. Notes Comput. Sci.* **2004**, *2690*, 1131–1135. [CrossRef]
37. Fu, D.; Guimaraes, G. Using Compression to Speed Up Image Classification in Artificial Neural Networks. Available online: https://www.danfu.org/files/CompressionImageClassification.pdf (accessed on 14 March 2023).
38. Arslan, H.S.; Archambault, S.; Bhatt, P.; Watanabe, K.; Cuevaz, J.; Le, P.; Miller, D.; Zhumatiy, V. Usage of compressed domain in fast frameworks. *Signal Image Video Process.* **2022**, *16*, 1763–1771. [CrossRef]
39. Hill, P.R.; Bull, D.R. Transform and Bitstream Domain Image Classification. *arXiv* **2021**, arXiv:2110.06740.
40. Tang, J.; Deng, C.; Huang, G.B.; Zhao, B. Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1174–1185. [CrossRef]
41. Hafed, Z.M.; Levine, M.D. Face Recognition Using the Discrete Cosine Transform. *Int. J. Comput. Vis.* **2001**, *43*, 167–188. [CrossRef]
42. Verma, V.; Agarwal, N.; Khanna, N. DCT-domain deep convolutional neural networks for multiple JPEG compression classification. *Signal Process. Image Commun.* **2018**, *67*, 22–33. [CrossRef]
43. Dong, Y.; Pan, W.D. Image Classification in JPEG Compression Domain for Malaria Infection Detection. *J. Imaging* **2022**, *8*, 129. [CrossRef]
44. Rajesh, B.; Dusa, N.; Javed, M.; Dubey, S.R.; Nagabhushan, P. T2CI-GAN: Text to Compressed Image generation using Generative Adversarial Network. *arXiv* **2022**, arXiv:2210.03734.
45. Li, X.; Zhang, Y.; Yuan, J.; Lu, H.; Zhu, Y. Discrete Cosin TransFormer: Image Modeling From Frequency Domain. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 5468–5478.
46. Chang, S.F.; Messerschmitt, D.G. A new approach to decoding and compositing motion-compensated DCT-based images. In Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA, 27–30 April 1993; IEEE: New York, NY, USA, 1993; Volume 5, pp. 421–424.
47. Merhav, N.; Bhaskaran, V. *A Fast Algorithm for Dct-Domain Inverse Motion Compensation*. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Atlanta, GA, USA, 7–10 May 1996; IEEE: New York, NY, USA; pp. 2307–2310
48. Meng, J.; Chang, S.F. CVEPS-a compressed video editing And parsing system. In Proceedings of the Forth International Conference on Multimedia, Boston, MA, USA, 18–22 November 1996; ACM: Rochester, NY, USA; pp. 43–53
49. Meng, J.; Chang, S.F. Tools for compressed-domain video indexing and editing. In Proceedings of the Storage and Retrieval for Still Image and Video Databases IV, San Jose, CA, USA, 28 January–2 February 1996; SPIE: Washington, DC, USA, 1996; Volume 2670, pp. 180–191.
50. Noguchi, Y.; Messerschmitt, D.G.; Chang, S.F. MPEG video compositing in the compressed domain. In Proceedings of the 1996 IEEE International Symposium on Circuits and Systems (ISCAS), Atlanta, GA, USA, 12–15 May 1996; IEEE: New York, NY, USA, 1996; Volume 2, pp. 596–599.
51. Smith, B.C.; Rowe, L.A. Compressed Domain Processing of JPEG-encoded images. *Real-Time Imaging* **1996**, *2*, 3–17. [CrossRef]
52. Kobla, V.; Doermann, D.S.; Lin, K.I. Archiving, indexing, and retrieval of video in the compressed domain. In *Multimedia Storage and Archiving Systems*; SPIE: Washington, DC, USA, 1996; Volume 2916, pp. 78–89. [CrossRef]
53. Kobla, V.; Doermann, D.S.; Lin, K.I.; Faloutsos, C. Compressed-domain video indexing techniques using DCT and motion vector information in MPEG video. In *Storage and Retrieval for Image and Video Databases V*; SPIE: Washington, DC, USA, 1997; Volume 3022, pp. 200–211.

54. Mandal, M.K.; Idris, F.; Panchanathan, S. A critical evaluation of image and video indexing techniques in the compressed domain. *Image Vis. Comput.* **1999**, *17*, 513–529. [CrossRef]
55. Wang, H.; Divakaran, A.; Vetro, A.; Chang, S.F.; Sun, H. Survey of compressed-domain features used in audio-visual indexing and analysis. *J. Vis. Commun. Image Represent.* **2003**, *14*, 150–183. [CrossRef]
56. Meng, J.; Chang, S.F. Embedding visible video watermarks in the compressed domain. In Proceedings of the 1998 International Conference on Image Processing, ICIP98 (Cat. No. 98CB36269), Chicago, IL, USA, 4–7 October 1998; IEEE: New York, NY, USA, 1998; Volume 1, pp. 474–477.
57. Nang, J.; Kwon, O.; Hong, S. Caption processing for MPEG video in MC-DCT compressed domain. In Proceedings of the Eighth ACM International Conference on Multimedia, Los Angeles, CA, USA, 30 October–3 November 2000; pp. 211–218.
58. Mansouri, A.; Aznaveh, A.M.; Torkamani-Azar, F.; Kurugollu, F. A low complexity video watermarking in H.264 compressed domain. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 649–657. [CrossRef]
59. Dutta, T.; Gupta, H.P. An efficient framework for compressed domain watermarking in p frames of high-efficiency video coding (HEVC)-encoded video. *ACM Trans. Multimed. Comput. Commun. Appl.* **2017**, *13*, 1–24. [CrossRef]
60. Acharya, S.; Smith, B. Compressed domain transcoding of MPEG. In Proceedings of the IEEE International Conference on Multimedia Computing and Systems (Cat. No. 98TB100241), Austin, TX, USA, 1 July 1998; IEEE: New York, NY, USA, 1998; pp. 295–304. [CrossRef]
61. Shanableh, T.; Ghanbari, M. Hybrid DCT/pixel domain architecture for heterogeneous video transcoding. *Signal Process. Image Commun.* **2003**, *18*, 601–620. [CrossRef]
62. Lin, H.Y.; Tsai, T.H.; Lin, Y.F. Video transcoder in DCT-domain spatial resolution reduction using low-complexity motion vector refinement algorithm. *Eurasip J. Adv. Signal Process.* **2008**, *2008*, 467290. [CrossRef]
63. Wee, S.J.; Vasudev, B. Compressed-domain reverse play of MPEG video streams. In *Multimedia Systems and Applications*; SPIE: Washington, DC, USA, 1999; Volume 3528, pp. 237–248.
64. Hesseler, W.; Eickeler, S. MPEG-2 compressed-domain algorithms for video analysis. *Eurasip J. Appl. Signal Process.* **2006**, *2006*, 056940. [CrossRef]
65. Alvar, S.R.; Bajić, I.V. MV-YOLO: Motion vector-aided tracking by semantic object detection. In Proceedings of the 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP), Vancouver, BC, Canada, 29–31 August 2018; IEEE: New York, NY, USA, 2018; pp. 1–5.
66. Ujiie, T.; Hiromoto, M.; Sato, T. Interpolation-based object detection using motion vectors for embedded real-time tracking systems. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 616–624.
67. Liu, Q.; Liu, B.; Wu, Y.; Li, W.; Yu, N. Real-time Online Multi-Object Tracking in Compressed Domain. *arXiv* **2022**, arXiv:2204.02081.
68. Chen, L.; Sun, H.; Katto, J.; Zeng, X.; Fan, Y. Fast Object Detection in HEVC Intra Compressed Domain. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021; IEEE: New York, NY, USA; 2021; pp. 756–760.
69. Alizadeh, M.; Sharifkhani, M. Compressed Domain Moving Object Detection Based on CRF. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 674–684. [CrossRef]
70. LAFFERTY, J. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Proc. 18th International Conference on Machine Learning, Williamstown, MA, USA, 28 June–1 July 2001; pp. 282–289.
71. Porikli, F.; Bashir, F.; Sun, H. Compressed domain video object segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 2–14. [CrossRef]
72. Tan, Z.; Liu, B.; Chu, Q.; Zhong, H.; Wu, Y.; Li, W.; Yu, N. Real Time Video Object Segmentation in Compressed Domain. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 175–188. [CrossRef]
73. Alvar, S.R.; Choi, H.; Bajic, I.V. Can you tell a face from a HEVC bitstream? In Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, USA, 10–12 April 2018; IEEE: New York, NY, USA, 2018; pp. 257–261.
74. Feng, J.; Li, S.; Li, X.; Wu, F.; Tian, Q.; Yang, M.H.; Ling, H. TapLab: A Fast Framework for Semantic Video Segmentation Tapping into Compressed-Domain Knowledge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1591–1603. [CrossRef] [PubMed]
75. Liu, Q.; Sung, A.H.; Qiao, M. Video steganalysis based on the expanded Markov and joint distribution on the transform domains - Detecting MSU stegovideo. In Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications, San Diego, CA, USA, 11–13 December 2008; pp. 671–674. [CrossRef]
76. Mstafa, R.J.; Elleithy, K.M. Compressed and raw video steganography techniques: A comprehensive survey and analysis. *Multimed. Tools Appl.* **2017**, *76*, 21749–21786. [CrossRef]
77. Muthuswamy, K.; Rajan, D. Salient motion detection in compressed domain. *IEEE Signal Process. Lett.* **2013**, *20*, 996–999. [CrossRef]
78. Fang, Y.; Lin, W.; Chen, Z.; Tsai, C.M.; Lin, C.W. A video saliency detection model in compressed domain. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 27–38. [CrossRef]
79. Zhu, S.; Liu, C.; Xu, Z. High-Definition Video Compression System Based on Perception Guidance of Salient Information of a Convolutional Neural Network and HEVC Compression Domain. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 1946–1959. [CrossRef]

80. Chadha, A.; Abbas, A.; Andreopoulos, Y. Compressed-domain video classification with deep neural networks: "There's way too much information to decode the matrix". In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: New York, NY, USA, 2017; pp. 1832–1836.
81. Zhu, W.; Yang, K.H.; Beacken, M.J. CIF-to-QCIF Video Bitstream Down-Conversion in the DCT Domain. *Bell Labs Tech. J.* **1998**, *3*, 21–29. [CrossRef]
82. Roma, N.; Sousa, L. Efficient hybrid DCT-domain algorithm for video spatial downscaling. *Eurasip J. Adv. Signal Process.* **2007**, *2007*, 057291. [CrossRef]
83. Zhang, J.; Li, S.; Kuo, C.C. Compressed-domain video retargeting. *IEEE Trans. Image Process.* **2014**, *23*, 797–809. [CrossRef]
84. Almeida, J.; Leite, N.J.; Torres, R.D.S. Online video summarization on compressed domain. *J. Vis. Commun. Image Represent.* **2013**, *24*, 729–738. [CrossRef]
85. Yamghani, A.R.; Zargari, F. Compressed Domain Video Abstraction Based on I-Frame of HEVC Coded Videos. *Circuits, Syst. Signal Process.* **2019**, *38*, 1695–1716. [CrossRef]
86. Basavarajaiah, M.; Sharma, P. Survey of compressed domain video summarization techniques. *ACM Comput. Surv.* **2019**, *52*, 1–29. [CrossRef]
87. Dorai, C.; Ratha, N.K.; Bolle, R.M. Detecting dynamic behavior in compressed fingerprint videos: Distortion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2000 (Cat. No. PR00662), Hilton Head, SC, USA, 13–15 June 2000; IEEE: New York, NY, USA, 2000; Volume 2, pp. 320–326. [CrossRef]
88. Arman, F.; Hsu, A.; Chiu, M.Y. Image processing on compressed data for large video databases. In Proceedings of the First ACM International Conference on Multimedia, Anaheim, CA, USA, 1–6 August 1993; pp. 267–272.
89. Darwish, A.M. A Video coprocessor: Video processing in the DCT domain. In Proceedings of the Media Processors, San Jose, CA, USA, 28–29 January 1999; SPIE: Washington, DC, USA, 1998; Volume 3655, pp. 158–168. [CrossRef]
90. Kaminsky, E.; Ginzburg, A.; Hadar, O. DCT-domain coder for digital video applications. *J. Real-Time Image Process.* **2010**, *5*, 259–274. [CrossRef]
91. Ilgin, H.A.; Chaparro, L.F. Low bit rate video coding using DCT-based fast decimation/interpolation and embedded zerotree coding. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 833–844. [CrossRef]
92. Shapiro, J.M. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Process.* **1993**, *41*, 3445–3462. [CrossRef]
93. Thies, W.; Hall, S.; Amarasinghe, S. *Manipulating Lossless Video in the Compressed Domain*; ACM: Rochester, NY, USA, 2009; p. 1166.
94. Mao, N.; Zhuo, L.; Zhang, J.; Li, X. *Fast Compression Domain Video Encryption Scheme for H.264/AVC Stream*; IEEE: New York, NY, USA, 2012.
95. Wang, Z.; Liu, X.; Feng, J.; Yang, J.; Xi, H. Compressed-Domain Highway Vehicle Counting by Spatial and Temporal Regression. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 263–274. [CrossRef]
96. He, P.; Li, H.; Wang, H.; Wang, S.; Jiang, X.; Zhang, R. Frame-Wise Detection of Double HEVC Compression by Learning Deep Spatio-Temporal Representations in Compression Domain. *IEEE Trans. Multimed.* **2021**, *23*, 3179–3192. [CrossRef]
97. Chen, P.; Yang, W.; Wang, M.; Sun, L.; Hu, K.; Wang, S. Compressed Domain Deep Video Super-Resolution. *IEEE Trans. Image Process.* **2021**, *30*, 7156–7169. [CrossRef]
98. Chen, J.; Ho, C.M. MM-ViT: Multi-modal video transformer for compressed video action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; IEEE: New York, NY, USA; pp. 1910–1921.
99. Patel, R.; Lad, K.; Patel, M. Study and investigation of video steganography over uncompressed and compressed domain: A comprehensive review. *Multimed. Syst.* **2021**, *27*, 985–1024. [CrossRef]
100. Mukhopadhyay, J. *Image and Video Processing in the Compressed Domain*; CRC Press: Boca Raton, FL, USA, 2011.
101. Babu, R.V.; Tom, M.; Wadekar, P. A survey on compressed domain video analysis techniques. *Multimed. Tools Appl.* **2016**, *75*, 1043–1078. [CrossRef]
102. Javed, M.; Nagabhushan, P.; Chaudhuri, B.B. A review on document image analysis techniques directly in the compressed domain. *Artif. Intell. Rev.* **2018**, *50*, 539–568. [CrossRef]