Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 3): pp. 4675-4684
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.03.479

Design And Development Of Mathematical Models For Image And Video Processing

Rakesh Chandra Bhadula Associate Professor, Department of Mathematics, Graphic Era Hill University, Dehradun Uttarakhand India.

Abstract

Applications like computer vision, multimedia, and remote sensing all heavily rely on the field of image and video processing. For analysing, modifying, and enhancing photos and movies, mathematical models are effective tools. The construction of mathematical models that are specifically suited for image and video processing jobs is examined in this research study. The paper outlines the fundamental ideas and procedures used to develop these models, emphasising their uses and benefits across a range of industries. It also includes case studies and experiments to show how well the suggested models work. The research develops and uses models in a methodical manner. In order to do this, it is necessary to identify the main issues with image and video processing, formulate mathematical models, and then rigorously assess these models using relevant benchmark datasets. The models are made to handle problems like video compression, object detection, motion estimation, noise reduction, and image enhancement. The suggested mathematical models integrate ideas from several branches of mathematics, including signal processing, linear algebra, optimisation, and statistical analysis. They use cutting-edge methods like deep learning, machine learning, and probabilistic modelling to conduct image and video processing tasks at the highest level of performance. Implemented and integrated into current image and video processing frameworks are the created mathematical models. The implementation includes designing effective algorithms, enhancing computational performance, and working with current hardware designs. The models are assessed quantitatively and qualitatively, and their effectiveness is contrasted with that of current state-of-the-art techniques.

I. Introduction:

The manipulation and analysis of visual data, including photos and movies, using a variety of methods and algorithms is referred to as "image and video processing." It entails the extraction of significant data, the improvement of the visual quality, and the accomplishment of tasks like object detection, tracking, segmentation, and compression. Numerous industries, including computer vision, multimedia, medical imaging, remote sensing, and surveillance, use image and video processing. Many different mathematical models, statistical methodologies, signal processing algorithms, and machine learning strategies are

used in image and video processing techniques. These techniques make it possible to analyse, manipulate, and comprehend visual data effectively, supporting applications in a variety of industries like entertainment, medicine, robotics, and surveillance systems. In order to effectively comprehend, analyse, and manipulate visual input, mathematical models play a critical role in image and video processing. Here are some main arguments supporting the necessity of mathematical models in this area:

Representation and Encoding: Mathematical models give picture and video data a methodical and succinct representation. They make it possible to efficiently store, transmit, and interpret massive amounts of data by enabling the compact and meaningful encoding of visual information. Mathematical models are useful for characterising noise and aberrations in picture and video data. Mathematical models make it easier to create algorithms for noise reduction and restoration by helping to understand its statistical features, which improves the clarity and quality of images.

Image enhancement: Mathematical models make it possible to improve pictures and movies by changing their contrast, brightness, and sharpness. Intelligently enhancing visual data with models based on statistical analysis, optimisation, or machine learning approaches can boost the perceptual quality overall and help with particular applications. Mathematical models give a framework for dividing up photos and videos into useful regions or objects for image segmentation and object recognition. They make it easier to recognise and extract elements, such colour, texture, and shape, which are crucial for tasks involving object detection and tracking. Models based on statistical analysis, graph theory, or clustering enable precise and effective segmentation and recognition.

Machine learning and Statistical Analysis: Mathematical models allow for the statistical analysis of picture and video data, revealing information about the distribution, correlations, and dependencies of the data. Tasks like anomaly detection, pattern recognition, and content-based retrieval can benefit from this knowledge. Deep neural networks and other machine learning models draw on mathematical underpinnings to learn intricate patterns and make predictions based on visual data. Mathematical models offer a quantitative foundation for assessing the performance of image and video processing algorithms. Performance evaluation and optimisation. Models enable comparison and the selection of the most efficient methods by defining objective measures and criteria. They also support the optimisation of algorithms for energy, memory, and computational efficiency.

II. Mathematical Fundamentals:

1. Signal processing techniques and transformations:

The analysis and manipulation of signals, including images and videos, depend fundamentally on the methods and transformations used in signal processing. Here are some

frequently used signal processing methods and transformations for the purpose of processing images and videos.

Convolution: A fundamental signal processing procedure, convolution is utilised for a number of functions, including filtering and feature extraction. Convolution is frequently carried out in image and video processing utilising a kernel or filter mask. The convolution procedure can be mathematically expressed as follows given an input signal f(x, y) and a filter h(x, y):

$$f(a, b) * h(x - a, y - b) = g(x, y) \quad (1)$$

where the summing is carried out over all values of a and b that are appropriate for the specified signal dimensions, and g(x, y) is the final output signal.

Discrete Cosine Transform (DCT): JPEG and other image and video compression methods frequently employ the discrete cosine transform. It changes a signal's spatial to frequency domain state. The equation: For a 2D signal f(x, y), the DCT can be calculated.

$$F(u, v) = f(x, y) * \cos((2x + 1)u/2N) * \cos((2y + 1)v/2N) = (u) * (v) \quad (2)$$

where F(u, v) denotes the representation of the frequency domain, N is the size of the signal, and (u) and (v) denote scaling factors.

Wavelet Transform: This adaptable method is employed for signal analysis and compression. It is appropriate for image and video processing applications since it records data in both the frequency and temporal domains. Mathematically, the 2D Continuous Wavelet Transform is stated as:

$$W(a, b) = *[(x - a)/s, (y - b)/s] = f(x, y)\, dx\, dy \quad (3)$$

Where f(x, y) is the input signal, is the wavelet function, and s is the scale parameter, W(a, b) represents the wavelet coefficients at scale a and position b.

2. Linear and nonlinear models for image and video representation

The pixel intensity-based model is a common linear mathematical representation for images. According to this paradigm, each image is represented as a matrix of pixel intensities, where each pixel value denotes the brightness or colour data for a particular area inside the image. For the sake of simplicity, let's look at a grayscale image where each pixel has a single intensity value that ranges from 0 to 255. The image can be represented as a matrix with the

symbol "I," where M stands for the number of rows (the image's height), and N stands for the number of columns (its width).

The pixel intensity-based model's linear representation of an image can be expressed mathematically as:

I = [I(1,1), I(1,2), ..., I(1,N);

I(2,1), I(2,2), ..., I(2,N);

...

I(M,1), I(M,2), ..., I(M,N)]

where I(i, j) denotes the pixel's intensity value in rows i and j.

With the help of matrices and linear algebraic methods, a variety of image processing activities, including filtering, enhancing, and transformation, may be carried out using this linear representation. It serves as the basis for numerous image processing methods and approaches, allowing for further image analysis and manipulation.

Because they are better able to describe complex relationships and structures seen in visual data, nonlinear models for representing images and videos have greater expressive potential. Here are two prevalent nonlinear models for representing images and videos.

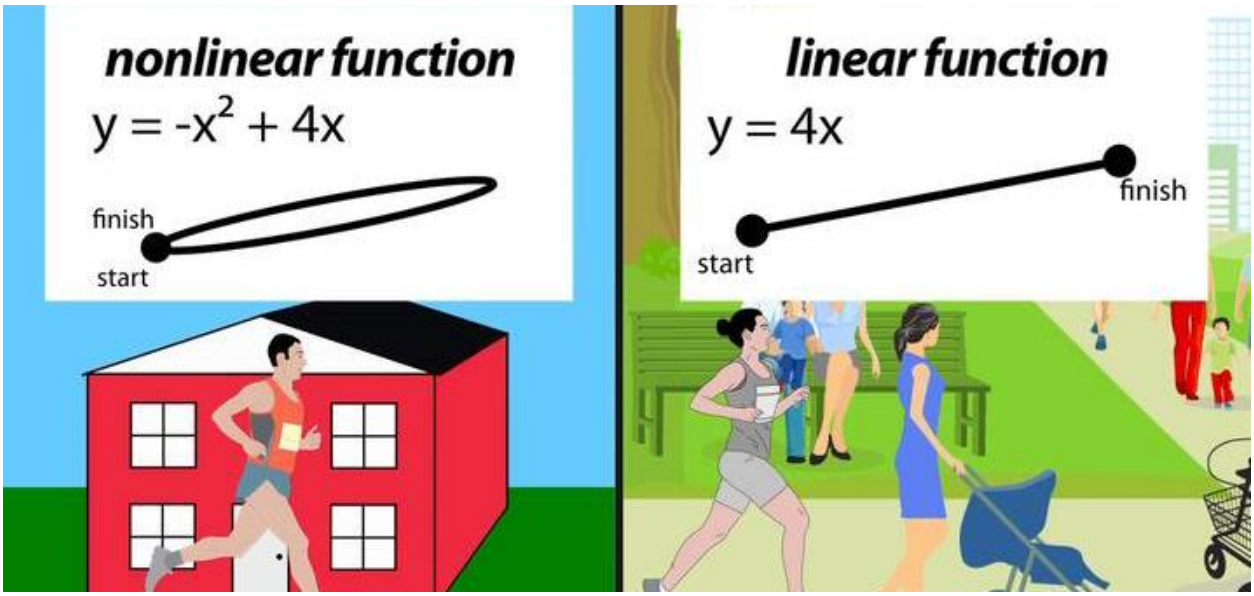

Figure 1: Linear Non linear function for Image via graph and table reprsentation

Autoencoders are neural network models that are capable of learning a compact representation of highly dimensional input by encoding and decoding it. In the processing of

images and videos, they are employed for nonlinear dimensionality reduction and feature extraction. The decoder network reconstructs the original data from the representation of the latent space, while the encoder network maps the input data to a lower-dimensional latent space.

The following equations can be used to model an auto encoder:

h = f_encoder(x) is the encoder.

X = f_decoder(h) is the decoder.

By capturing hierarchical and nonlinear relationships in visual data, Deep Neural Networks (DNNs) have revolutionised the representation of images and videos. DNNs can learn very nonlinear transformations since they have numerous layers of interconnected neurons.

Convolutional neural networks (CNNs) are frequently employed for representing images and videos. Convolutional layers are used by CNNs to extract spatial information from image or video frames, while fully connected layers are used for more complex representation and classification tasks.

The following equations can be used to represent a typical CNN layer:

Layer of Convolution: z = f_conv(W * x + b)

Layer that is fully connected: h = f_fc(W * x + b

In this case, x stands for the input data (such as an image or video frame), W stands for the learnable weights, b for the biases, and z and h stand for, respectively, the output activations of the convolutional and fully connected layers. The activation functions (such as ReLU and sigmoid) applied element-by-element to the input are represented by f_conv and f_fc.

Deep neural networks may capture increasingly complex and abstract representations of image and video input by stacking numerous convolutional and fully connected layers. By capturing complex patterns and relationships that may be difficult for linear models to handle, these nonlinear models, autoencoders, and deep neural networks enable effective image and video representation. They have substantially aided the development of projects like generative modelling, object identification, object categorization, and video interpretation.

III. Efficient Modeling

We propose to keep only the instructive frequency components from each Discrete Cosine Transform (DCT) map Di for effective image modelling, which is analogous to the famous JPEG compression, which achieves compression by discarding non-visually significant values

through quantization [2]. We investigated three different compression techniques in this study:

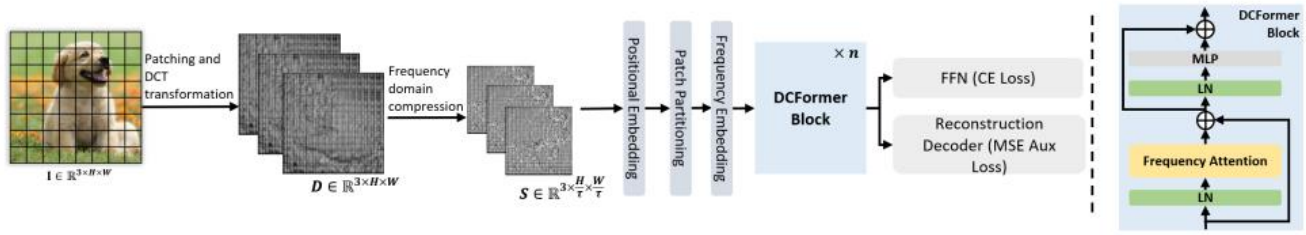

Figure 2: Overview of Model for image restoration

Thresholding-based Compression:

Another compression technique is thresholding, which eliminates frequency components below a predetermined threshold. By setting a threshold value and only keeping the DCT coefficients that are higher than it, we may identify the components that are visually noteworthy. We lessen the amount of data required to depict the image by removing the low magnitude coefficients.

$$\mathrm{Qi} = \mathrm{Di} * (|\mathrm{Di}| > \mathrm{Th})$$

where |Di| denotes the absolute value of Di, Qi denotes the thresholded DCT coefficients, Di denotes the original DCT coefficients, Th denotes the threshold value.

Based on Sparse Representation Compression:

A sparse set of basis functions or dictionary atoms is used to represent the image in a sparse representation. To find a sparse representation of the DCT coefficients, we employ methods like compressed sensing and sparse coding in this compression scheme. We can rebuild the image with high fidelity while using less data by choosing a small number of important frequency components.

$$\mathrm{Qi} = \big|\big|\mathrm{Di} - \mathrm{Dk} * \mathrm{Xi}\big|\big| + \big|\big|\mathrm{Xi}\big|\big|0$$

where ||.|| denotes a norm (such as the L0 or L1 norm), Qi denotes the sparse representation of DCT coefficients, Di denotes the original DCT coefficients, Dk denotes dictionary atoms, Xi denotes the sparse coefficients, and denotes the sparsity level.

The suggested model as shown in figure 1 uses a DCT-based frequency representation and works with RGB images as input. A frequency compression module is an optional addition that offers a significant boost in efficiency at the expense of a marginally lower level of performance. The positional and frequency embeddings are added to the frequency-based

representation after which it is put through a sequence of DCFormer blocks. The frequency attention method is made to work with several transformer attentions, including neighbour attention and SWIN attention. For classification tasks, a linear projection with cross-entropy (CE) loss is used, and when frequency compression is used, an auxiliary mean squared error (MSE) reconstruction loss can be used.

Average:

An average pooling over the DCT map using the kernel

$$Q(i,j) = \text{round}\left(\frac{D(i,j)}{T}\right) * T$$

The quantization step size is T, the quantization coefficient at location (i, j) is D(i, j), the original DCT coefficient is Q(i, j), and the round() function rounds the result to the nearest integer.

Soft-selection:

A cross-attention based soft-selection method on Di as:

$$Si = \text{MHCA}\,(\text{Conv}\ 2D(Di), Qemb$$

Hard-selection: We concentrate on maintaining the low-frequency components while eliminating higher frequency components in order to execute a hard selection of frequency components in the zigzag pattern. A popular scanning order for moving through a matrix's DCT coefficients is the zigzag pattern.

$$\mathbf{Q(i,j) = 0, if\ (i + j) > K\ D(i,j), otherwise}$$

Compression Ratio:

A comparison of the trade-offs between accuracy and efficiency was possible thanks to the evaluation of the classification performance under various frequency compression ratios. A higher compression ratio, such as = 4, led to reduced precision due to the loss of more information, but also to fewer Floating Point Operations (FLOPs). A lower compression ratio, on the other hand, produced the opposite result. The ablation study led to the choice of = 0.5 because it provided the optimal balance between effectiveness and efficiency. It was notable to note that the DCFormer with no frequency down-sampling (S = 1) obtained the same accuracy and FLOPs as SWIN-T with RGB picture input. This finding implies that the RGB representation and the frequency domain representation are equally effective.

DCT Patch Size:

Investigated was the effect of applying various DCT patch sizes on photos of various resolutions. The results imply that the image resolution affects the DCT patch size selection. Smaller DCT patch sizes, like 8x8, typically perform better on smaller images (such as 256x256 and 384x384). This is because a smaller patch size is adequate to gather the necessary frequency information yet smaller images have less details and variances.

However, performance is not reliably improved by merely raising the input resolution with the same DCT patch size. This is due to the limited information that small DCT patches with small DCT bases can hold. Larger DCT patches are more appropriate as image resolution rises because they can transmit more frequency information and capture finer details.

Compression method:

It was discovered during the investigation that average pooling, which is frequently used for spatial down-sampling, significantly reduced performance when applied to the frequency domain. This decline in performance is caused by the fact that it makes no sense to average data points from multiple frequency bands because they do not directly correlate with one another in space. The loss of frequency data due to average pooling might lead to the absence of crucial information needed for precise frequency domain modelling. In order to learn a weighted average-based compression, cross-attention was investigated as a solution to this problem. Cross-attention application, however, necessitates the addition of convolution layers to the input DCT map, which increases computational complexity. This goes against the purpose of producing effective compression because it necessitates more computation and reduces effectiveness.

VII. Conclusion

In many applications, the creation and improvement of mathematical models for image and video processing is essential. These models enable operations like compression, transformation, augmentation, and analysis by effectively allowing us to extract and manipulate visual information. A formal foundation for representing and processing images and videos is provided by mathematical models, which enables us to take advantage of mathematical ideas and methods to provide effective and precise outcomes. They offer a means of describing the underlying patterns, relationships, and structures seen in visual data. For the encoding and compression of images, linear mathematical models like the Discrete Cosine Transform (DCT) provide a solid foundation. They take advantage of the relationship between pixels and frequencies to enable effective coding and storage of image and video data. Contrarily, nonlinear models offer greater adaptability and sophisticated capabilities for the representation of images and videos. Tasks like picture identification, object detection, and video segmentation are made possible by these models' ability to record complicated relationships and nonlinear transformations. Deep learning architectures like convolutional neural networks (CNNs) and recurrent neural networks

(RNNs) are examples of nonlinear models. The mathematical models selected will depend on the particular specifications of the current image or video processing task. It is necessary to take into account variables like computational effectiveness, precision, compression ratio, and visual quality. The models are made even more successful by using the right transformations, such as filtering, edge detection, and feature extraction. There is ongoing study into the creation and advancement of mathematical models for the processing of images and videos. The current focus is on investigating novel designs, enhancing current methods, and improving performance. In order to accomplish more complex and intelligent processing, researchers are also looking into ways to add semantic data, spatial-temporal linkages, and context awareness into the models.

Reference:

[1] Venkateshwar,R., Patil,P., Naveen, A., Muthukumar,V., «Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based Hardware Descriptive Language,» International Journal of Theoretical and Applied Computer Sciences, 2006.

[2] W. J. MacLean, «An Evaluation of the Suitability of FPGAs for Embedded Vision Systems,» chez Computer vision and pattern recognition, San Diego, CA, 2005.

[3] RG, Handel-C Language Reference Manual, Celoxica Limited, 2005.

[4] S. Partners, "SysML Specification v. 1.0a," 2005. [Online]. Available: http://www.sysml.org.

[5] Tim Schattkowsky, Jan Hendrik Hausmann, Gregor Engels, «Using UML Activities for System-On-Chip design and synthesis,» Springer, 2006.

[6] C.T.Johnston, D.G.Bailey, P.Lyons, «A Visual Environment for Real-Time Image Processing in Hardware (VERTIPH),» EURASIP Journal on Embedded Systems, p. 1–8, 2006.

[7] M.AIT ALI, M.ELEULDJ, «An intermediate modelisation Language for embedded system developemnt chain COCODEL : Un Langage intermédiaire de modélisation pour une chaine de développement des systèmes embarqués COCODEL,» chez JODIC'2012, Rabat, 2012.

[8] E. Planas, J. Cabot et a. C. Gómez, «Verifying Action Semantics Specifications in UML,» Springer-Verlag Berlin Heidelberg, vol. CAiSE 2009, n° %1LNCS 5565, pp. 125- 140, 2009.

[9] OMG, OMG Unified Modeling Language Specifications (Action Semantics), OMG, 2009.

[10] J.-C. Fernandez, G. Hubert , A. Kerbrat, L. Mounier , R. Mateescu et M. Sighireanu, «CADP : A Protocol Validation and Verification Toolbox,» chez CAV '96 : Proceedings of the 8th International Confe- rence on Computer Aided Verification, London, UK, 2006.

[11] E. Foundation, «Graphical Modeling Framework,» The Eclipse Foundation, [En ligne]. Available: http://www.eclipse.org/modeling/gmp/.

[12] E. Foundation, «Acceleo,» Eclipse Foundation, [En ligne]. Available: http://www.eclipse.org/acceleo/.

[13] RG, Pixel Streams manual, Celoxica, 2005.

[14] Charest, L., Aboulhamid, E.M., «A VHDL/SystemC Comparison in Handling Design Reuse,» 2008.

[15] OMG, OMG Unified Modeling LanguageTM (OMG UML), Superstructure, Version 2.3, 2010.

[16] OMG, "Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification _ Version 1.1," Object Management Group, 2011.