March 21, 2019

Task-specific color spaces and compression for machine-based object recognition

Shumeet Baluja

David Marwood

Nicholas Johnston

Task-specific color spaces and compression for machine-based object recognition

ABSTRACT

An image is compressed to reduce the memory or bandwidth it occupies. Compression is presently carried out such that the reconstructed (decompressed) image is faithful to the original image. In some recent contexts, images are generated that are not necessarily intended for human viewership. For example, such images are generated for the purposes of machine-based tasks such as action-detection, scene-recognition, etc. In such cases, compression that is driven by fidelity of the decompressed image to the original can be sub-optimal. This disclosure describes techniques to compress images based on the end use of the image. For example, if an image is used for the purposes of detecting particular objects within it, then image compression is driven by an object detector. Portions of the image that are irrelevant to detecting the sought objects are excised during compression. The result is a more efficient, task-specific, encoding of the image.

KEYWORDS

- Image compression
- Synthetic image
- Training image
- Machine learning
- Machine vision
- Object recognition
- Task-specific compression
- Task-specific color space
- Color space

BACKGROUND

In many contexts, a majority of machine-based tasks on images, e.g., object recognition, scene detection, action detection, image segmentation, etc., are done on a server, rather than on a client. This is because of the following reasons:

- Neural networks that are used for object detection/recognition can be large and compute-intensive.
- Client versions of these neural networks (e.g., for mobile devices) do not have accuracy comparable to the server-side networks. Mobile devices, while capable of capturing images, often do not have extensive on-board computation capabilities.

When an image or video, e.g., captured on a mobile device, is compressed for sending to a server, it is usually compressed using standard techniques, e.g., to a format such as jpeg, webp, etc. These techniques are optimized for fidelity of the reconstructed (decompressed) image to the original, as judged by human aesthetic criteria. However, images sent to task-oriented servers often have no requirement for being viewable by a human; indeed, such images may never be seen by a human. They are only processed by a neural network for the purposes of fulfilling certain tasks, e.g., object detection (e.g., to recognize objects in the image, such as car, dog, bottle of soda, etc.), scene recognition, action detection, etc. Consequently, compressing an image such that the decompressed image is faithful to the original image is a sub-optimal strategy to save memory and bandwidth when the end use of the image is as input to machine-based tasks.

DESCRIPTION

This disclosure describes compression and color space techniques tailored to machine-based tasks, e.g., object recognition, scene detection, etc. Per the techniques, aspects of the image

that are irrelevant to the task are excised regardless of their aesthetic appeal to humans. For example, if the machine-based task is to recognize objects in an image, then portions of the image that do not pertain to objects are not encoded. The result is a more efficient, task-specific, encoding of the image.

Two approaches are described for task-specific image compression. The goal of each approach is to reduce the size of the image that is transmitted from mobile device to server. In a first approach, the image is cast into a color space that is discretized into a small number of bits on the mobile device. In a second approach, a neural network is trained to compress images in a manner similar to standard compression; however, the training signal is optimized for the specific task on hand, e.g., object recognition, etc.

Approach 1: shifted, specialized, color spaces with discretization

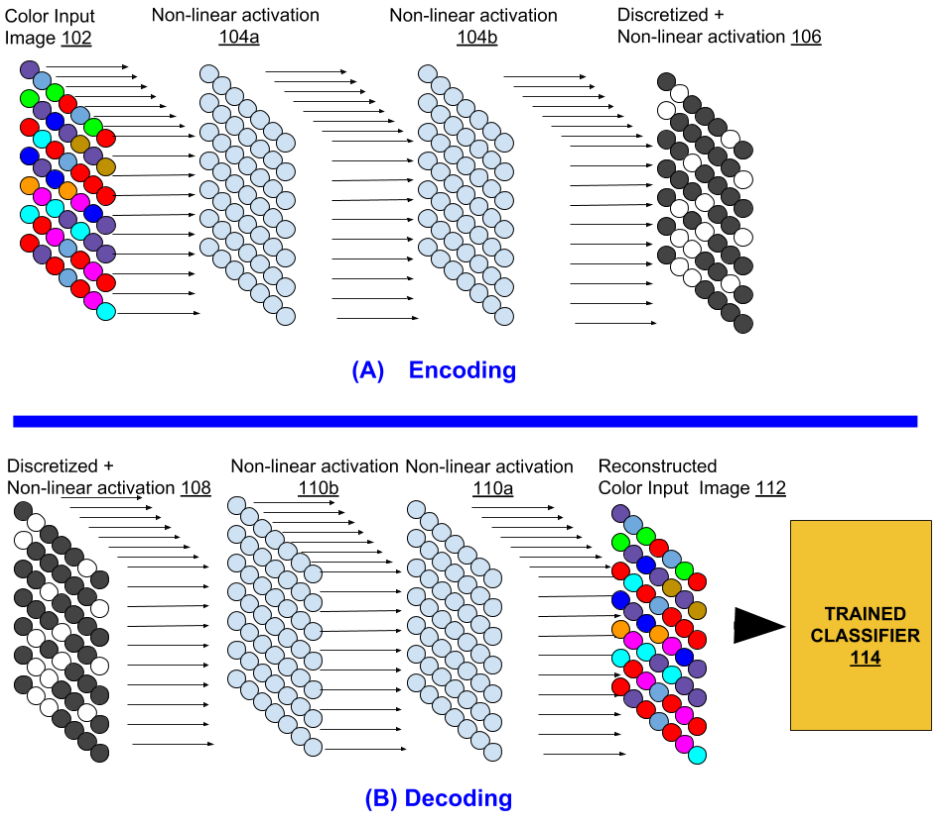

Fig. 1: Task-specific compression (encoding) and reconstruction (decoding) of images

Fig. 1A illustrates task-specific compression, also known as encoding and Fig. 1B illustrates reconstruction, also known as decoding, of images using shifted, specialized color spaces with discretization. Encoding, which is typically performed on a mobile device, is performed as follows. A color image (102) is received as input, and a discretized output (106) is computed by passing the image through one or more neural networks that serve as non-linear activation stages (104a-b). The non-linear activation stages transform the colors. The discretization shown here is of two levels with a single depth channel. Multiple, e.g., four, levels can be used with three channels, analogous to RGB channels.

Contrary to conventional encoders, connections between layers of the encoder are not fully connected, nor do they use small, convolutional patches across the entirety of the image. Rather, the connections here are one-to-one. Alternatively, they can be thought of as size (1, 1) convolutions with stride (1, 1). Edge-like information can be sent with larger, e.g., 3x3, convolution sizes.

The encoding network of Fig. 1A is a simple network with a low total number of connections, since each input pixel is connected to a single unit in the previous layer, albeit in full depth. The original input image can be of relatively small size, e.g., 200x200 pixels. The last layer of the network uses a discretizing activation [1]. The discretization turns the floating point input to the unit into one-of-N values, where N can be a small number, e.g., 4, 8, 16, etc.

Once propagated to the final layer, the discretized image is sent to task-specific servers for the purposes of decoding. The number of bits for each channel being small, a high compression ratio can be achieved.

Decoding of the image (Fig. 1B) is done by performing the encoding steps in reverse. A discretized image (108) received by the server is fed into one or more non-linear activation

networks (110b-a). A high-performance network, even of nearly arbitrary complexity, can reconstruct the original color image (112), which is fed into a trained classifier (114), or other task-specific unit, e.g., action detector, image segmenter, etc. The main computation is done at the server, which can accommodate large, task-specific, neural networks, e.g., with many millions of connections. Once the tasks are complete, e.g., objects are recognized, the results of e.g., the detected object and/or its bounding box, are sent back to the device.

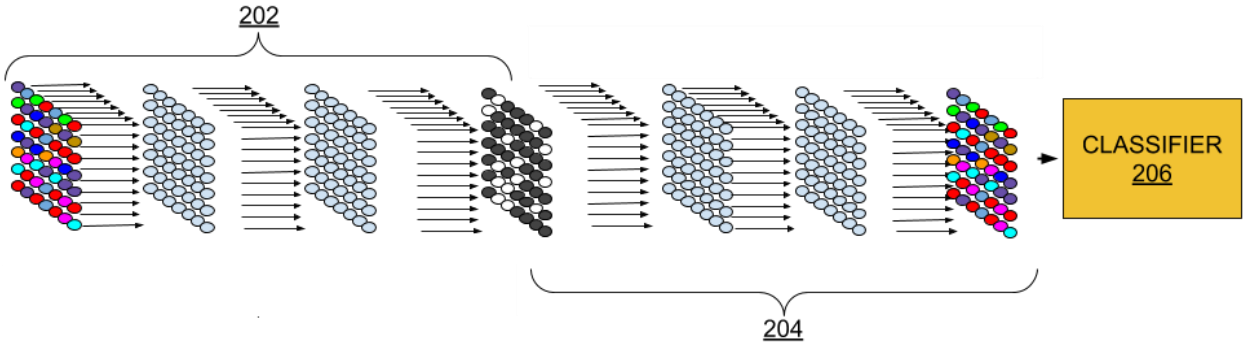

Fig. 2: Training task-specific image compression networks

Fig. 2 illustrates the training of task-specific image compression networks, per techniques of this disclosure. An encoder (202) and a decoder (204) are trained together, guided by an error signal emanating from the classifier or other task-specific unit (206). As mentioned earlier, minimization of distortion, e.g., using L1 or L2 metrics, between the original and reconstructed images is *not* a criterion for optimizing the networks. Rather, it is the ability of the classifier to correctly classify the image that serves as an error signal to train the networks. The neural network on a mobile device is relatively lightweight, and is a small fraction of the computation required for machine recognition. The task of machine recognition is performed by the trained network (206).

Although Fig. 2 illustrates the encoder and decoder as being of the same size, there is no requirement for their dimensions to be similar. The classifier is already fully trained, e.g., there is

no retraining of the classifier; rather, it is the classifier that is used to train the encoder-decoder network. Each component of the encoder-decoder network is trained simultaneously, with the classification error signal passing through the entire network, e.g., through the levels of decoder to the levels of the encoder. Based on the error (training signal) provided by the classifier, the encoder-decoder network rejects or retains to appropriate resolution selected portions and colors of the image that are relevant to image reconstruction.

Approach 2: direct training of image-compression neural networks to achieve task-specific goals

For mobile devices that are not compute-limited, instead of discretizing color spaces, images are compressed in a manner similar to standard compression. Image compression is performed not with the goal of faithful reconstructability, but with the goal of accurately accomplishing certain machine-based tasks, e.g., scene classification, object detection, action recognition, image segmentation, etc. The machine-based task unit, e.g., scene classifier, generates an error signal that is propagated backwards through the image compression network. As mentioned earlier, such an error signal is *not* a conventional error such as a sum-of-squares error, L1-error, etc.

The architectural difference between the two approaches is that in the second approach, the networks need not be restricted to 1x1 connections. Instead, arbitrarily complex networks, e.g., fully connected or convolutional networks, can be used. The 1x1 connections of the first approach ensure that the RGB value of each pixel is transformed non-linearly, but does not take into consideration the RGB value of its neighbors. By using convolutional networks, which are more suitable for image compression, this restriction is lifted [2, 5, 6, 7, 8].

In the manner, the described techniques compress images by using an error metric that measures the quality of the compressed image in terms of its efficacy for a machine-based task.

The tasks can include a variety of machine-based image understanding and processing tasks, e.g., object detection, scene classification, action recognition, image segmentation, etc.

CONCLUSION

This disclosure describes techniques to compress images based on the end use of the image. For example, if an image is used for the purposes of detecting particular objects within it, then image compression is driven by an object detector. Portions of the image that are irrelevant to detecting the sought objects are excised during compression. The result is a more efficient, task-specific, encoding of the image. Tasks that can drive image compression include a variety of machine-based image understanding and processing tasks, e.g., object detection, scene classification, action recognition, image segmentation, etc.

REFERENCES

[1] Baluja, Shumeet, David Marwood, Michele Covell, and Nick Johnston. "No Multiplication? No Floating Point? No Problem! Training Networks for Efficient Inference." *arXiv preprint arXiv*:1809.09244 (2018).

[2] Toderici, George, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. "Variable rate image compression with recurrent neural networks." *arXiv preprint arXiv:*1511.06085 (2015).

[3] Liu, Zihao, Tao Liu, Wujie Wen, Lei Jiang, Jie Xu, Yanzhi Wang, and Gang Quan. "DeepN-JPEG: a deep neural network favorable JPEG-based image compression framework." In *Proceedings of the 55th Annual Design Automation Conference*, p. 18. ACM, 2018.

[4] Chinchali, Sandeep P., Eyal Cidon, Evgenya Pergament, Tianshu Chu, and Sachin Katti. "Neural Networks Meet Physical Networks: Distributed Inference Between Edge Devices and

the Cloud." *In Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, pp. 50-56. ACM, 2018.

[5] Toderici, George, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. "Full resolution image compression with recurrent neural networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5306-5314. 2017.

[6] Theis, Lucas, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. "Lossy image compression with compressive autoencoders." arXiv preprint arXiv:1703.00395 (2017).

[7] Dony, Robert D., and Simon Haykin. "Neural network approaches to image compression." Proceedings of the IEEE 83, no. 2 (1995): 288-303.

[8] Baldi, Pierre. "Autoencoders, unsupervised learning, and deep architectures." In Proceedings of ICML workshop on unsupervised and transfer learning, pp. 37-49. 2012.