A Video Codec Designed for AI Analysis

Originally published at https://www.unite.ai/a-video-codec-designed-for-ai-analysis/

Though techno-thriller *The Circle* (2017) is more a comment on the ethical implications of social networks than the practicalities of external video analytics, the improbably tiny 'SeeChange' camera at the center of the plot is what truly pushes the movie into the 'science-fiction' category.

The 'SeeChange' camera/surveillance device from techno-thriller 'The Circle' (2017).

A wireless and free-roaming device about the size of a large marble, it's not the lack of solar panels or the inefficiency of drawing power from other ambient sources (such as radio waves) that makes SeeChange an unlikely prospect, but the fact that it's going to have to compress video 24/7, on whatever scant charge it's able to maintain.

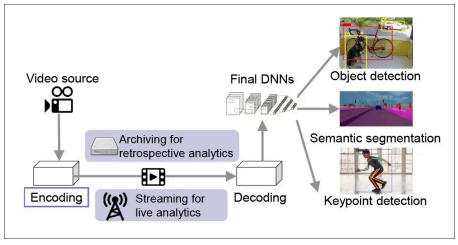
Powering cheap sensors of this type is a core area of research in computer vision (CV) and video analytics, particularly in non-urban environments where the sensor will have to eke out the maximum performance from very limited power resources (batteries, solar, etc.).

In cases where such an edge IoT/CV device of this type must send image content to a central server (often through conventional cell coverage networks), the choices are hard: either the device needs to run some kind of lightweight neural network locally in order to send only *optimized* segments of relevant data for server side processing; or it has to send 'dumb' video for the plugged-in cloud resources to evaluate.

Though motion-activation through event-based Smart Vision Sensors (SVS) can cut down this overhead, that activation monitoring also costs energy.

Clinging to Power

Furthermore, even with infrequent activation (i.e. a sheep occasionally wanders into view), the device doesn't have sufficient power to send gigabytes of uncompressed video; neither does it have enough power to constantly run popular video compression codecs such as H.264/5, which are expecting hardware that's either plugged in or not far from the next charging session.



Video analytics pipelines for three typical computer vision tasks. The video encoding architecture needs to be trained for the task at hand, and usually for the neural network that will receive the data. Source: https://arxiv.org/pdf/2204.12534.pdf

Though the widely diffused H.264 codec has lower energy consumption than its successor H.265, it has poor compression efficiency. Its successor, H.265, has better compression efficiency, but higher power consumption. While Google's open source VP9 codec beats them both in each area, it requires higher local computation resources, which presents additional problems in a supposedly cheap IoT sensor.

As for analyzing the stream locally: by the time you've run even the lightest local neural network in order to determine which frames (or areas of a frame) are worth sending to the server, you've often spent the power you would have saved by just sending all the frames.

Extracting masked representations of cattle with a sensor that's unlikely to be grid-connected. Does it spend its limited power capacity on local semantic segmentation with a lightweight neural network; by sending limited information to a server for further instructions (introducing latency); or by sending 'dumb' data (wasting energy on bandwidth)? Source: https://arxiv.org/pdf/1807.01972.pdf

Its clear that 'in the wild' computer vision projects need dedicated video compression codecs that are optimized to the requirements of specific neural networks across specific and diverse tasks such as semantic segmentation, keypoint detection (human movement analysis) and object detection, among other possible end uses.

If you can get the perfect trade-off between video compression efficiency and minimal data transmission, you're a step nearer the SeeChange, and the ability to deploy affordable sensor networks in unfriendly environments.

AccMPEG

New research from the University of Chicago might have taken a step nearer to such a codec, in the form of AccMPEG – a novel video encoding and streaming framework that operates at low latency, high accuracy for server-side Deep Neural Networks (DNNs), and which has remarkably low local compute requirements.

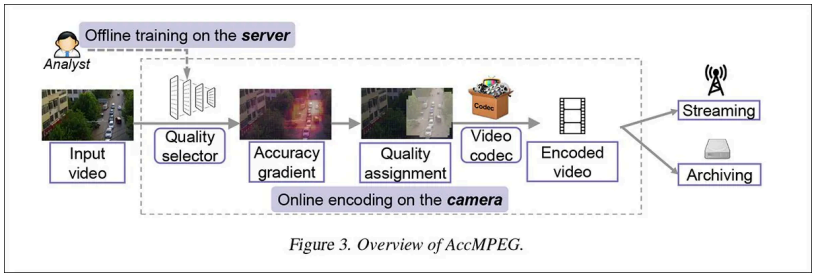


Figure 3. Overview of AccMPEG.

Architecture of AccMPEG. Source: https://arxiv.org/pdf/2204.12534.pdf

The system is able to make economies over prior methods by assessing the extent to which each 16x16px macroblock is likely to affect accuracy of the server-side DNN. Previous methods have, instead, generally had to assess this kind of accuracy based on each pixel in an image or else to perform electrically expensive local operations to assess which regions of the image might be of most interest.

In AccMPEG, This accuracy is estimated in a custom module called AccGrad, which measures the ways in which the encoding quality of the macroblock is likely to be pertinent to the end usage case, such as a server-side DNN that's trying to count people, perform skeleton estimation on human movement, or other common computer vision tasks.

As a frame of video arrives into the system, AccMPEG initially processes it through a cheap quality selector model, titled *AccModel*. Any areas which are not likely to contribute to the useful calculations of a server-side DNN are essentially ballast, and should be marked for encoding at the lowest possible quality, in contrast to salient regions, which should be sent at better quality.

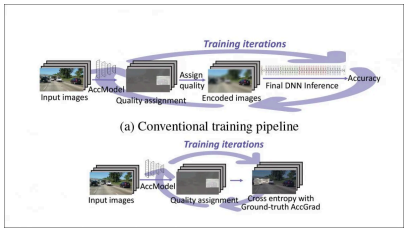
This process presents three challenges: can the process be performed quickly enough to achieve acceptable latency without using energy-draining local compute resources? Can an optimal relationship between frame-rate and quality be established? And can a model be quickly trained for an individual server-side DNN?

Training Logistics

Ideally, a computer vision codec would be pre-trained on plugged-in systems to the exact requirements of a specific neural network. The AccGrad module, however, can be directly derived from a DNN with only two forward propagations, at a saving of ten times the standard overhead.

AccMPEG trains AccGrad for a mere 15 epochs of three propagations each through the final DNN, and can potentially be retrained 'live' using its current model state as a template, at least for similarly-specced CV tasks.

AccModel uses the pretrained MobileNet-SSD feature extractor, common in affordable edge devices. At a turnover of 12 GFLOPS, the model uses only a third of typical ResNet18 approaches. Besides batch normalization and activation, the architecture consists only of convolutional layers, and its compute overhead is proportional to the frame size.



AccGrad removes the need for final DNN inference, improving deployment logistics.

Frame Rate

The architecture runs optimally at 10fps, which would make it suitable for purposes such as agricultural monitoring, building degradation surveillance, high-view traffic analysis and representative skeleton inference in human movement; however, very fast-moving scenarios, such as low-view traffic (of cars or people), and other situations in which high frame rates are beneficial, are unsuited to this approach.

Part of the method's frugality lies in the premise that adjacent macroblocks are likely to be of similar value, up until the point where a macroblock falls below estimated accuracy. The areas obtained by this approach are more clearly delineated, and can be calculated at greater speed.

Performance Improvement

The researchers tested the system on a $60 Jetson Nano board with a single 128-core Maxwell GPU, and various other cheap equivalents. OpenVINO was used to offset some of the energy requirements of the very sparse local DNNs to CPUs.

AccModel itself was originally trained offline on a server with 8 GeForce RTX 2080S GPUs. Though this is a formidable array of computing power for an initial model build, the lightweight retraining that the system makes possible, and the way that a model can be adjusted to certain tolerance parameters across different DNNs that are attacking similar tasks, means that AccMPEG can form part of a system that needs minimal attendance in the wild.

First published 1st May 2022.

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai