Building a Video Annotation Platform Part 3

Common video data challenges and solutions

ALEGION

Building A Video Annotation Platform Part 3

Creating a video-based computer vision (CV) application is hard. Video annotation is hard. After spending countless amounts of time and money, data science teams often find that ***errors or corruption in their ground truth video data has created a spiral of problems throughout their model training process.***

In this final part in our series, **Building a Video Annotation Platform** we explore common challenges specific to video data, explain how these issues can affect downstream model accuracy, and show how Alegion's video annotation (VA) solution can help prevent any costly surprises due to issues with ground truth video files.

Part 3 Outline:

- Quality annotations start with pristine video
- Encoders v decoders
- The problem with decoders and video annotation
- The challenges of video capture devices and encoding
- How decoders cope with degraded data
- What does it all mean? Most VA platforms handle issues by forcing pre-processing. Alegion doesn't.
- How to increase speed, quality, and accuracy of video

Quality Annotations Start With Pristine Video

As we've worked with customers building video-based CV models over the years, it's been clear that computer vision teams are often caught off-guard by underlying issues with their ground truth video files.

After spending huge amounts of time, money, and compute overhead to process their video data, these teams are brought up short by inconsistencies between annotation data and the original video data, a problem that is just as serious as inaccurate annotations.

When we were building Alegion Control, we knew we needed to help CV teams validate video quality *first*, in order to deliver the high **quality, fast, accurate annotation data** they need.

Encoders vs. Decoders

All video data goes through a process of compression and uncompression as it moves from collection point to playback; this basic process is called encoding and decoding. It's worth clarifying what we mean by this before we tackle specific video data errors and corruption.

Modern video compression algorithms have many provisions that compensate for issues when capturing and playing back a video stream. This robustness allows them to work well in many scenarios, but it is equally good at masking problems because:

- Encoders do their very best to keep encoding, regardless of the condition of the network or stream.

- Decoders know how to deal with errors and keep on playing even when corruption occurs or data is missing. One of their main jobs is to HIDE any issues with a stream, and they do this in any number of ways.

ENCODER

Takes a raw video stream and creates a compressed video bitstream using a given compression algorithm. The encoder usually sits on the edge device and compresses the stream to ease transmission and storage.

DECODER

Uncompresses a compressed video bitstream in order to be displayed. Any application that can read a video file does so using a decoder (e.g. your web browser, ffmpeg, video editing software). The decoder must support the same standard used by the encoder, but does not need to be written by the same provider of the encoder (spoiler alert: this causes problems).

Decoding and Video Annotation

This tension between the encoding standard and the multiplicity of decoding options creates big problems when it comes to video annotation.

Decoders can choose to handle underlying issues with the stream differently. For example, playing back a clip missing an I-frame may look different in Chrome than it does in Safari due to a difference in the decoders used by different browsers. This presents a challenge when trying to annotate a video in a browser because what is seen by the user making the annotations isn't necessarily what a downstream decoder may provide to the training dataset creation process.

Unless you are one of the few with pristine video sources, understanding the underlying decoding process for your processing pipeline is key to ensuring that your labels match up with your annotations.

The Challenges With Video Capture Devices and Encoding

The majority of our customers rely on streaming IP cameras to power their solutions. These cameras are small, low cost, and only require a network connection to stream video to a centralized location. IP cameras are ubiquitous - it's common to see them in retail settings, along roadways, construction sites, etc. Most of these systems aren't originally built for computer vision applications, but they can be used without much complication for the most part.

The objective function of these IP cameras is to ensure continuous streaming even under poor network conditions. When a network bandwidth issue or hiccup occurs, there are many different ways a system may be implemented to deal with this degraded condition. Some systems will duplicate frames to maintain a constant frame rate while others will drop frames or reduce the encoding bitrate.

All of these failure modes can be problematic when it comes to data labeling.

How Decoders Cope With Degraded Video Data

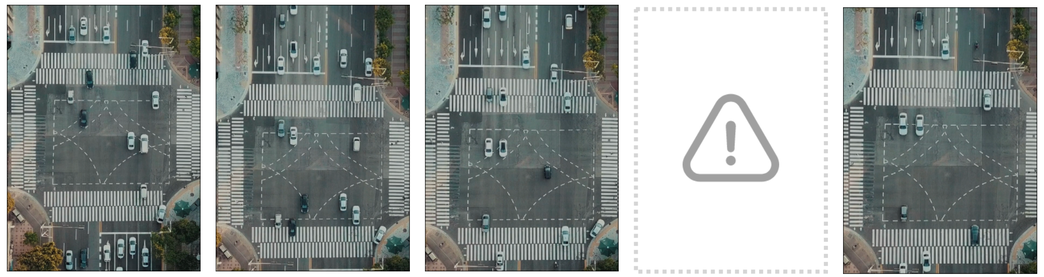
Let’s discuss two of the most common issues with video data and how decoders (and subsequently annotators) cope with them.

Missing frames

Missing frames are commonly caused by lost packets due to network interruptions while streaming from the camera to the centralized collection point.

When encountering missing frames, the decoder may duplicate previous frames, or show black frames to maintain a constant frame rate (thus temporarily adding new content that did not exist in the source video), or simply skip ahead to the next frame.

If frames are duplicated, this causes problems during labeling because ***the annotator is working on a frame that doesn't exist in the original video and is not preserved after the annotation session.*** This is most problematic when you get further into the dataset serialization process, and ***you end up with more annotated frame data than actual frames***. This scenario will cause annotations to be out of sync with the image data which in the end has the same negative effect as really inaccurate annotations.



How Decoders Cope With Degraded Video Data

Detecting missing frames

Most compressed video formats rely on a **presentation timestamp** (PTS) metadata field in the bitstream to keep audio and video synchronized as well as to support variable frame rates. In short, there is a *pts_time* for each frame that tells the decoder exactly when it should be displayed.

One of the methods that can be used to detect missing frames using this information is to analyze the *pts_time* for inconsistent intervals. Below is an example where there is a missing frame between frames #4-5 (the pkt_pts_time jumps by 67ms instead of 33ms):

	pkt_pts	pkt_pts_time	pkt_dts	pkt_dts_time	best_effort_timestamp	best_effort_timestamp_time
0	0	0.000000 s	0	0.000000 s	0	0.000000 s
1	1	0.000011 s	1	0.000011 s	1	0.000011 s
2	3060	0.034000 s	2	0.000022 s	3060	0.034000 s
3	6030	0.067000 s	3060	0.034000 s	6030	0.067000 s
4	9000	0.100000 s	6030	0.067000 s	9000	0.100000 s
5	15030	0.167000 s	15030	0.167000 s	15030	0.167000 s
6	18090	0.201000 s	18090	0.201000 s	18090	0.201000 s
7	21060	0.234000 s	21060	0.234000 s	21060	0.234000 s
8	24030	0.267000 s	24030	0.267000 s	24030	0.267000 s

Row number represents the frame number. pts_time should increment by ~33ms (this is a 29.97fps video).

If not detected, this could cause an offset of one frame once the frames are exported leading to inaccurate annotations. During normal playback on your desktop or in a browser, these types of issues are not reported, they're simply handled by the decoder.

Fixing missing frames

Once detected, fixing individual clips requires a straightforward re-encoding step. After re-encoding, you'll still be missing frames, but the video will be internally consistent and the number (and order) of annotated frames will match the new re-encoded reference video. The video is essentially **“re-striped"** meaning every frame will have a new *pts_time* with the expected interval.

Missing frames aren't a big deal as long as you are able to identify when they occur and re-stripe the video.

How Decoders Cope With Degraded Video Data

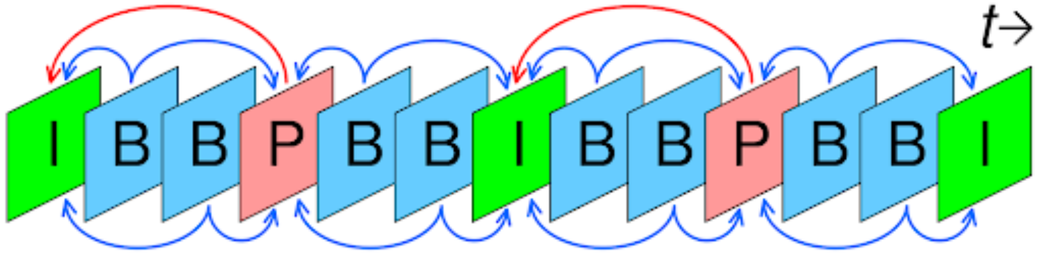
Compression artifacts

Lossy compression algorithms typically rely on a GOP (group of pictures) mechanism in order to reduce the data required to represent the video. There are three major picture types in video compression:

1. **I−frames** (Intra-coded picture): higher fidelity I-frames can essentially stand alone as "full" frames, what we would think of as a complete image.

2. **P−frames** (Predicted picture): P-frames store interpolated data from the previous I or P-frames, each new P-frame is essentially just the changes in the image from the previous frame.

3. **B−frames** (Bidirectional predicted picture): B-frames also use intra-frame compression, but are informed by both previous and future I-frames and P-frames.



By Cmglee - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=64558505

How Decoders Cope With Degraded Video Data

Problems with compression artifacts

The important thing to understand is that an error in one frame can cause artifacts in the entire GOP, essentially the frames between I-frame boundaries. Unfortunately, it's very common to see portions of a clip contain encoder induced compression artifacts that are attributable to corruption instead of bitrate or color-depth-induced quantization.

The decoder wants to do its very best to playback the video so it will make up its own visual data to display. Again, different decoders may display this generated data differently.

Normal video on the left side, visible P-frame corruption on the right side. Observe color and shape irregularities, increased posterization.

What Does It All Mean? Most VA Platforms Handle Issues By Forcing Pre-processing. Alegion Doesn't.

Many competing VA platforms force a pre-processing step for video data in order to address issues with ground truth video files.

This forced pre-processing step is problematic for three big reasons:

1 **It's time-consuming.**

Choosing to treat the video as sequences of images or forcing a pre-processing step that transcodes the video is fine when dealing with a small number of short videos, but adds a HUGE amount of time, complexity, and compute overhead when working at production scale.

2 **You lose quality.**

Any time a video is re-encoded (transcoded), there will be a loss in quality. When this step is avoided, image quality is preserved. Additionally, most customers want to control the encoding profile, and many competing video annotation systems do not provide this flexibility.

3 **You risk mismatched data.**

On other systems, if your videos are pre-processed by their system and you are not given a new reference video along with your labels, you are not guaranteed that the annotations will match up with your original videos.

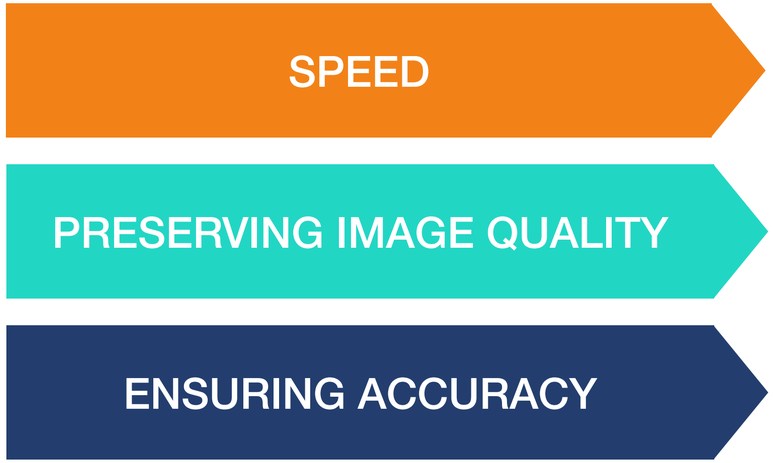
How Alegion Handles Issues With Video Data

Identifying and remediating these types of issues with your ground truth video files is not straightforward.

To ease the pain, Alegion provides a video validation tool that can be integrated into your processing pipeline that not only identifies these issues (and many more) but handles the creation of specific remediation commands to ensure your video is pristine *before beginning annotation*.

In the vast majority of cases, annotation can proceed without any significant remediation; in the few cases where video files need more clean-up, our video validation tool identifies and addresses the issues.

This approach provides several benefits:



By using Alegion's video validation tool before annotation begins, you can avoid forced pre-processing and rest assured that your downstream training process won't be affected.

Summary

We hope this short series has provided you with some insight into the technical and engineering challenges that come with building a world class, scalable, and highly responsive video annotation solution.

We're proud to work with some of the biggest innovators in the space as they build their next-generation video-based vision platforms.

If you have questions or want to learn more, ***please reach out to solutions@alegion.com or get in touch.***

ALEGION CONTROL

See for yourself! Sign up for Alegion Control today and your first 150 annotation hours are on us!

Want to learn more about Alegion's self service annotation platform?

Reach out to solutions@alegion.com

Follow us on

ALEGION