September 20, 2017

Non-deterministic video frame sampling to thwart frame insertion attacks

Juhyun Lee

Non-deterministic video frame sampling to thwart frame insertion attacks

ABSTRACT

Accurate machine understanding of videos is important, e.g., to maintain the integrity and policy compliance of videos at a video hosting website. For example, a video-hosting website with a policy of hosting only family-friendly videos deploys video understanding systems to automatically exclude non-compliant videos. Techniques of image analysis are applied to individual video frames to understand the video. Due to the computational cost of processing individual frames, videos are sampled, e.g., at a rate of one frame per second, and only the sampled frames are subjected to image analysis.

Sampling-based video understanding is susceptible to attack. For example, such systems fail to detect a non-compliant video, if the video includes policy-compliant frames inserted to match sampling instants. This disclosure utilizes randomization of sampling instants to thwart frame insertion attacks that attempt to mask actual video content. Randomized sampling for video understanding also assures reproducibility such that the understanding of the video is independent of the sampling instants.

KEYWORDS

- Video understanding
- Subsampling attack
- Pseudorandom sampling
- Anti-abuse computing

BACKGROUND

Machine perception technologies, e.g., deep neural nets, are used to automatically understand unstructured data such as images. Extending machine perception to video, frames of a video are analyzed to understand the entire video. However, neural-net inference is computationally expensive. Hence video is sub-sampled, e.g., at one frame per second, and inference is run only on the sampled frames to understand the video.

Attacks have been developed that can fool sampling-based video-understanding systems, e.g., where the inference from the sampled frames does not match the actual content of the video. An example attack is to determine sampling instants, e.g., by trial-and-error, and to replace sampled video frames with other images. Such an attack could deceive spam and abuse systems that leverage video content signals. For example, if the inserted images that are sampled comply with spam/ abuse standards, a neural-net can conclude that the video meets the standards, even when the video retains spam/ abusive content. In another example, such an attack can promote videos unfairly, e.g., by injecting popular entities into irrelevant videos and click-baiting users, or by fooling discovery systems into promoting irrelevant videos.

Robustness to sampling attacks is important for automatic video understanding technologies. Further, another desirable property for such technologies is reproducibility, e.g., the understanding of video content should be independent of sampling instants.

DESCRIPTION

Techniques of this disclosure thwart the sampling attack by randomizing the sampling instant. A video sampled at some nominal sampling rate, e.g., one frame per second, is not sampled exactly at each integral second. Rather, the video is sampled at a pseudo-random time instant within an interval around the nominal sampling instants. For example, the sampling instant may be within ±25% of the period around the nominal sampling instants.

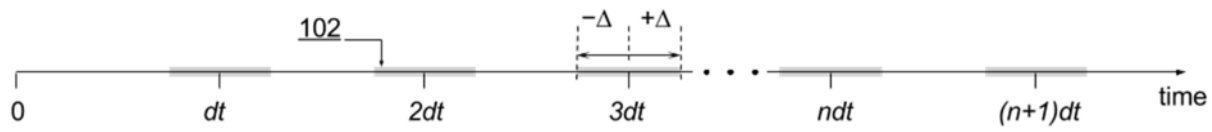

Fig. 1: Randomized sampling instants to thwart sampling attack

Fig. 1 illustrates the randomization of sampling instants per techniques of this disclosure. Time-points marked at integral multiples of *dt*, e.g., 0, *dt*, *2dt*, ..., represent the nominal sampling instants. For example, *dt*=1 represents a nominal sampling rate of one frame per second. The grey regions (102) around each nominal sampling instant represent the times within which actual sampling of frames takes place. The width of the grey region is represented as 2Δ. For example, if Δ=0.25 seconds, actual sampling is performed within ±0.25 seconds of the nominal sampling instant. The initial sampling instant, shown as zero time in Fig. 1, is itself pseudo-randomly selected as an instant between the start of the video and *dt* seconds into the video. The time-line of Fig. 1 is drawn with reference to the pseudo-randomly selected initial sampling instant.

```
202:  T  = frame rate                    # set T to desired frame rate after sampling

204:  dt = 1/T                           # dt is the interval between samples

206:  hash_code = code unique to video
      seed = hash_code encrypted with an internal private crypto key
      randomizer.initialize(seed)        # uniform distribution

208:  desired_t = randomizer.next() * dt# first sampling point can be any time
                                         # between 0 and dt

210:  v = video frame among v_x (0 ≤ x ≤ n-1) whose timestamp is closest to desired_t

212:  sampled_video_frames.append(v)

214:  while true:
         desired_t += 3 * dt / 4 + randomizer.next() * dt / 2
                                         # the next timestamp is anywhere between
                                         # 3dt/4 and 5dt/4
         if desired_t > m * dt:          # mdt is such that it is just less than
             break                       # t_{n-1} the end of the video

         sampled_video_frames.append(v)

216:  Create W from video frames in sampled_video_frames
```

Fig. 2: Pseudocode for randomized frame sampling

Fig. 2 illustrates pseudocode for randomized frame sampling of a video, per techniques of this disclosure. *V* is a video file containing video frames v_0, v_1, ..., v_{n-1} with presentation timestamps t_0, t_1, ..., t_{n-1}, respectively. A sub-sampling of *V* is sought at a positive frame rate *T*, and a new video *W* is obtained using the sub-sampled frames.

First, the frame rate *T* is set (202). The interval between samples is established as the reciprocal of the frame rate (204). A randomizer is seeded with a seed that is unique for the video. The seed is derived from video content or metadata, e.g., an identifier for the video (206). The selection of the seed is such that a malicious user would find it computationally prohibitive to establish a map between sampled frames and the video or metadata. The selection of the seed is derived such that it is difficult for an uploader of a video to specify it or otherwise manipulate its setting. The seed can be based on one or more of the following: video content

bytes, metadata such as a video identifier assigned by a video hosting service, one or more user identifiers, video header block, video length, resolution, etc. The seed is uniquely associated with the video after sanitization, e.g., after stripping metadata that can be manipulated by a user. The distribution generated by the randomizer is uniform between 0 and 1.

An initial sampling instant, *desired_t*, is pseudo-randomly and uniformly selected between 0 and *dt* (208). An initial video frame *v* is selected from those video frames v_x ($0 \leq x \leq n-1$) whose timestamp is closest to *desired_t* (210). The initial video frame *v* is appended to a set *sampled_video_frames* (212). A loop runs through the sequence of video frames selecting sampling instants *desired_t* and video frames *v* that are within a Δ of successive nominal sampling instants (214).

In Fig. 2, Δ is, for example, 25% of *dt*, such that the next sampling instant is a uniform random instant between $3dt/4$ and $5dt/4$. The loop exits (illustrated with a *break* statement) if the number of frames in the video is exhausted, e.g., if *desired_t* equals or exceeds the length of the video. A new video *W* comprising the sampled video frames is obtained (216).

The performance of the pseudo-random sampling based video-understanding system, as described herein, is not different from a video-understanding system that uses fixed sampling instants. This is demonstrated by comparing the precision-versus-recall curves of video-understanding systems based on fixed sampling and pseudo-random sampling instants.

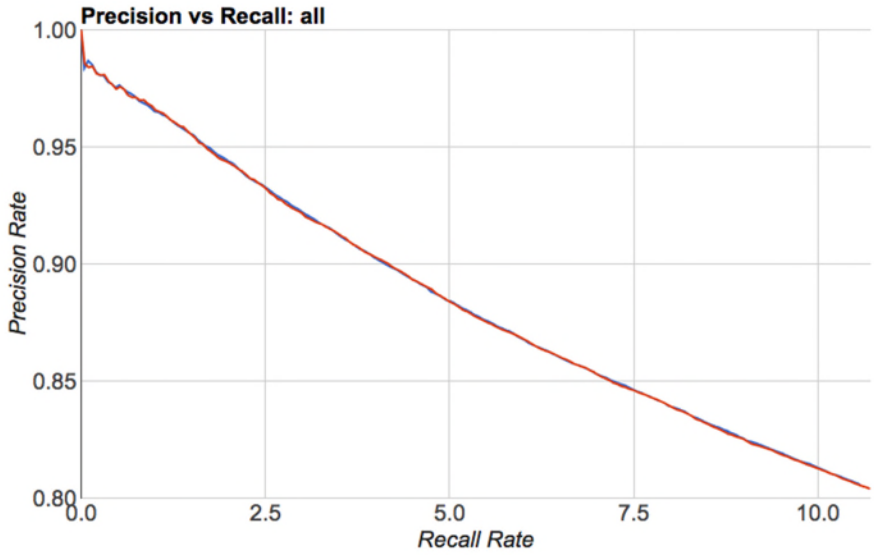

Fig. 3: Precision versus recall for video-understanding systems based on fixed sampling versus pseudo-random sampling

Fig. 3 shows the precision-versus-recall performance of a video-understanding system based on pseudo-random sampling (blue) and fixed sampling (red). It is evident that there is no performance difference attributable to pseudo-random sampling of video frames.

As an alternate to the system described herein, frame sampling can also be randomized by pseudo-randomly selecting a sampling instant and keeping it fixed for blocks of several frame-lengths, then pseudo-randomly changing it for a next block of several frame-lengths, and so on. Further, sampling rates can be changed between blocks.

CONCLUSION

Techniques of this disclosure counter frame-sampling prediction by adversaries by deploying a robust sampling scheme for video understanding. The described sampling scheme pseudo-randomly samples a video while maintaining adequate time-intervals between sampled frames. Adversaries that attempt to mislead video understanding systems, e.g., to bypass policy guidelines of a hosting website, by deliberate insertion of frames are thwarted.