ature is in a pre-release state and might change or have limited support. For more information, see the product launc
s (/products/#product-launch-stages).

This page describes how to annotate a video stream from standard live streaming protocols.

The Video Intelligence API Streaming API enables real-time streaming analysis for live media. Supported features include:

- Live Label Detection
- Live Shot Change Detection
- Live Explicit Content Detection
- Live Object Detection and Tracking

The AIStreamer (https://github.com/google/aistreamer/tree/master/ingestion) ingestion library provides a set of open source interfaces and example code to connect to the Video Intelligence API Streaming API. The library supports:

- HTTP Live Streaming (HLS): an HTTP based media streaming and communication protocol.
- Real Time Streaming Protocol (RTSP): a network control protocol for streaming media servers. It is used in conjunction with Real Time Protocol (RTP) and Real Time Control Protocol (RTCP).
- Real Time Messaging Protocol (RTMP): a protocol for streaming audio, video, and data over the Internet.

The AIStreamer ingestion library includes the following examples (incuding a Docker example).

- Live Streaming (/video-intelligence/docs/streaming/live-streaming): Instructions for supporting live streaming protocols (HLS, RTSP and RTMP) in Video Intelligence API.
- Docker & Kubernetes (/video-intelligence/docs/streaming/docker-kubernetes): Instructions for using our docker example and kubernetes deployment.

- Live Label Detection (/video-intelligence/docs/streaming/live-label-detection): Instructions for streaming label analysis.
- Live Shot Change Detection (/video-intelligence/docs/streaming/live-shot-change-detection): Instructions for streaming shot change analysis.
- Live Explicit Content Detection (/video-intelligence/docs/streaming/live-explicit-content): Instructions for streaming explicit content analysis.
- Live Object Detection and Tracking (/video-intelligence/docs/streaming/live-object-tracking): Instructions for streaming object detection and tracking analysis.

The AIStreamer ingestion library includes the following three directories:

- client (https://github.com/google/aistreamer/tree/master/ingestion/client): Python & C++ client libraries for connecting to Video Intelligence.
- env (https://github.com/google/aistreamer/tree/master/ingestion/env): Docker example for AIStreamer ingestion.
- proto (https://github.com/google/aistreamer/tree/master/ingestion/proto): Proto definitions and gRPC interface for Video Intelligence.

The open source AIStreamer ingestion library is based on the following Google-owned and third-party open source libraries.

- Bazel (https://bazel.build/): A build and test tool with multi-language support.
- gRPC (https://grpc.io/): A high performance, open-source universal RPC framework.
- Protobuf (https://developers.google.com/protocol-buffers): Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data.
- rules_protobuf (https://github.com/pubref/rules_protobuf): Bazel rules for building protocol buffers and gRPC services.
- glog (https://github.com/google/glog): C++ implementation of the Google logging module.

- gflags (https://github.com/gflags/gflags): C++ library that implements command-line flags processing.
- ffmpeg (https://www.ffmpeg.org/): A complete, cross-platform solution to record, convert and stream audio and video.
- gStreamer (https://gstreamer.freedesktop.org/): Another cross-platform multimedia processing and streaming framework.