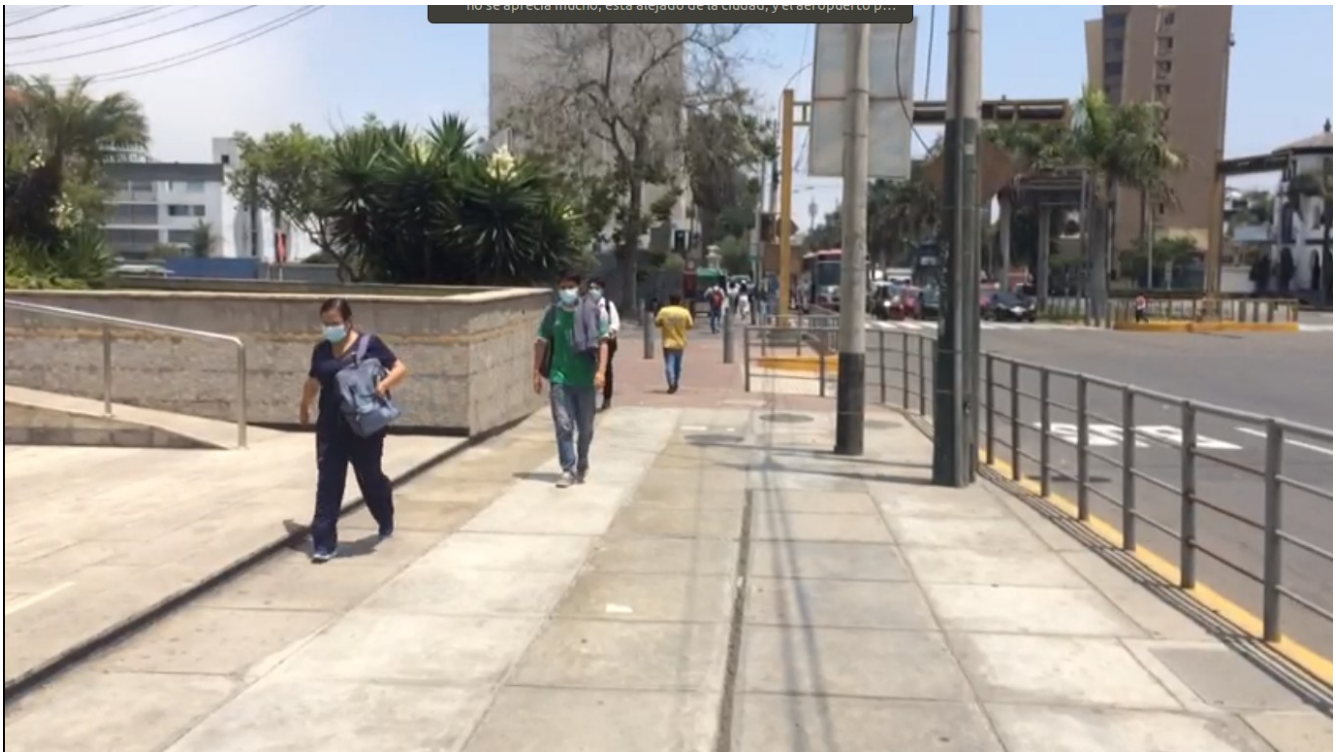
Etiquetando de Video Javier Prado



```
from typing import Optional, Sequence
from google.cloud import videointelligence_v1 as vi
def detect_labels(
        video_uri: str,
        mode: vi.LabelDetectionMode,
        segments: Optional[Sequence[vi.VideoSegment]] = None,
) -> vi.VideoAnnotationResults:
        video_client = vi.VideoIntelligenceServiceClient()
        features = [vi.Feature.LABEL_DETECTION]
        config = vi.LabelDetectionConfig(label_detection_mode=mode)
        context = vi.VideoContext(segments=segments, label_detection_config=config)
        request = vi.AnnotateVideoRequest(
                input_uri=video_uri,
                features=features,
                video_context=context,
        )
        print(f'Processing video "{video_uri}"...')
        operation = video_client.annotate_video(request)
        return operation.result().annotation_results[0]  # Single video
```

```
from datetime import timedelta
video_uri = "gs://cloud-samples-data/video/JaneGoodall.mp4"
mode = vi.LabelDetectionMode.SHOT_MODE
segment = vi.VideoSegment(
    start_time_offset=timedelta(seconds=0),
    end_time_offset=timedelta(seconds=37),
)
results = detect_labels(video_uri, mode, [segment])


def print_video_labels(results: vi.VideoAnnotationResults):
    labels = results.segment_label_annotations
    sort_by_first_segment_confidence(labels)
    print(f" Video labels: {len(labels)} ".center(80, "-"))
    for label in labels:
        categories = category_entities_to_str(label.category_entities)
        for segment in label.segments:
            confidence = segment.confidence
            t1 = segment.segment.start_time_offset.total_seconds()
            t2 = segment.segment.end_time_offset.total_seconds()
            print(
                f"{confidence:4.0%}",
                f"{t1:7.3f}",
                f"{t2:7.3f}",
                f"{label.entity.description}{categories}",
                sep=" | ",
            )
def sort_by_first_segment_confidence(labels: Sequence[vi.LabelAnnotation]):
    labels.sort(key=lambda label: label.segments[0].confidence, reverse=True)
def category_entities_to_str(category_entities: Sequence[vi.Entity]) -> str:
    if not category_entities:
        return ""
    entities = ", ".join([e.description for e in category_entities])
    return f" ({entities})"



print_video_labels(results)
```

n [17]: print_video_labels(results)

```
------------------------------ Video labels: 18 ------------------------------
 95% |   0.000 |   9.343 | street (road)
 89% |   0.000 |   9.343 | urban area (city)
 88% |   0.000 |   9.343 | vehicle
 85% |   0.000 |   9.343 | sidewalk (city)
 84% |   0.000 |   9.343 | road
 83% |   0.000 |   9.343 | public space (city)
 80% |   0.000 |   9.343 | pedestrian (person)
 73% |   0.000 |   9.343 | neighbourhood (geographical feature)
 57% |   0.000 |   9.343 | walkway
 56% |   0.000 |   9.343 | town (geographical feature)
 51% |   0.000 |   9.343 | lane (road)
 38% |   0.000 |   9.343 | residential area (geographical feature)
 38% |   0.000 |   9.343 | car (vehicle)
 37% |   0.000 |   9.343 | boardwalk (walkway)
 36% |   0.000 |   9.343 | transport
 35% |   0.000 |   9.343 | infrastructure
 33% |   0.000 |   9.343 | walking (sports)
 30% |   0.000 |   9.343 | land vehicle (vehicle)
```

In [18]: