# Mobile Vision API - concatenate new detector object to continue frame processing

Asked 9 years, 3 months ago Modified 5 years, 7 months ago

Viewed 20k times Part of Mobile Development Collective

19

I want to use the new face detection feature that the vision API provides along with additional frame processing in an application. For this, I need to have access to the camera frame that was processed by the face detector, and concatenate a processor using the face detected data.

As I see in the sample, the CameraSource abstracts the detection and camera access, and I can't have access to the frame being processed. Are there examples of how to get the camera frame in this API, or, maybe, create and concatenate a detector that receives it? Is that possible at least?

Thanks, Lucio

MD android google-vision

Share

Improve this question

asked Aug 30, 2015 at 18:47

Lxu

437 ● 1 ● 5 ● 12

Follow

## 3 Answers

Sorted by: Highest score (default)

23

Yes, it is possible. You'd need to create your own subclass of Detector which wraps FaceDetector and executes your extra frame processing code in the detect method. It would look something like this:

```
class MyFaceDetector extends Detector<Face> {
  private Detector<Face> mDelegate;

  MyFaceDetector(Detector<Face> delegate) {
    mDelegate = delegate;
  }

  public SparseArray<Face> detect(Frame frame) {
    // *** add your custom frame processing code here
    return mDelegate.detect(frame);
  }

  public boolean isOperational() {
    return mDelegate.isOperational();
  }

  public boolean setFocus(int id) {
    return mDelegate.setFocus(id);
  }
}
```

You'd wrap the face detector with your class, and pass your class into the camera source. It would look something like this:

```
    FaceDetector faceDetector = new FaceDetector.Build
            .build();
```

```
    MyFaceDetector myFaceDetector = new MyFaceDetector

    myFaceDetector.setProcessor(/* include your proces

    mCameraSource = new CameraSource.Builder(context,
            .build();
```

Your detector will be called first with the raw frame data.

Note that the image may not be upright, if the device is rotated. You can get the orientation through the frame's metadata.getRotation method.

One word of caution: once the detect method returns, you should not access the frame pixel data. Since the camera source recycles image buffers, the contents of the frame object will be eventually overridden once the method returns.

EDIT: (additional notes) You could also avoid the boilerplate code of `MyFaceDetector` using a MultiDetector like this:

```
MultiDetector multiDetector = new MultiDetector.Builde
    .add(new FaceDetector.Builder(context)
                .build())
    .add(new YourReallyOwnDetector())
    .build();
```

Also note the use of `FaceTrackerFactory` in conjuction with `MultiProcessor` described there.

Share Improve this answer edited Sep 6, 2016 at 23:17

Follow

Rafael T
15.7k • 15 • 84 • 148

answered Aug 31, 2015 at 14:54

pm0733464
2,862 • 15 • 16

No idea how this can still allow you to crop the image. The frame comes in and you can edit it but Frame is private and there is no setBitMap... – JPM Jul 25, 2016 at 20:31

@JPM you will probably have to create a new Frame using Frame.Builder() and set the cropped bitmap into it using setBitmap(Bitmap image) or setImageData(ByteBuffer data, int width, int height, int format). All the params can be obtained directly from the input frame or the frame metadata. – user1689757 Jan 11, 2017 at 3:00

22

Here's the final solution I settled on. It assumes the box is centered on the screen.

```
public class BoxDetector extends Detector {
    private Detector mDelegate;
    private int mBoxWidth, mBoxHeight;

    public BoxDetector(Detector delegate, int boxWidth
        mDelegate = delegate;
        mBoxWidth = boxWidth;
        mBoxHeight = boxHeight;
    }

    public SparseArray detect(Frame frame) {
        int width = frame.getMetadata().getWidth();
        int height = frame.getMetadata().getHeight();
        int right = (width / 2) + (mBoxHeight / 2);
        int left = (width / 2) - (mBoxHeight / 2);
        int bottom = (height / 2) + (mBoxWidth / 2);
```

```
        int top = (height / 2) - (mBoxWidth / 2);

        YuvImage yuvImage = new YuvImage(frame.getGray
ImageFormat.NV21, width, height, null);
        ByteArrayOutputStream byteArrayOutputStream =
ByteArrayOutputStream();
        yuvImage.compressToJpeg(new Rect(left, top, ri
byteArrayOutputStream);
        byte[] jpegArray = byteArrayOutputStream.toByt
        Bitmap bitmap = BitmapFactory.decodeByteArray(
jpegArray.length);

        Frame croppedFrame =
                new Frame.Builder()
                        .setBitmap(bitmap)
                        .setRotation(frame.getMetadata
                        .build();

        return mDelegate.detect(croppedFrame);
    }

    public boolean isOperational() {
        return mDelegate.isOperational();
    }

    public boolean setFocus(int id) {
        return mDelegate.setFocus(id);
    }
}
```

Wrap this class in your detector like this

```
BarcodeDetector barcodeDetector = new BarcodeDetector.
BoxDetector boxDetector = new BoxDetector(barcodeDetec
```

Share Improve this answer edited May 11, 2017 at 19:37

Follow

answered May 11, 2017 at 12:13



MCR

**1,643** ● 3 ● 21 ● 37

---

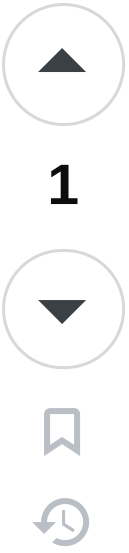
@saturov how should I use detect function? What frame should I pass? I've created BoxDetector: `new BoxDetector(barcodeDetector, cameraView.getHeight()/3, cameraView.getWidth());` and when I'm calling boxDetector.detect I don' know what frame to pass. – mikro098 Jul 5, 2017 at 13:45

1 @MCR how use boxDetector ? – upward Jan 31, 2018 at 6:52

In case someone else runs into this, I was getting `IllegalArgumentException: rectangle is not inside the image` The solution was `override fun receiveFrame(frame: Frame?) = delegate.receiveFrame(frame)` – Kraiden May 8, 2018 at 0:09

how to detect only face box using Sparsearray? – DoctorWho Sep 3, 2018 at 9:57

Thank you, I was searching solution for the same, and it worked for me. Although I tried for creating bitmap using the OverLayView and CameraSourcePreiview but nothing worked. – Kotdroid Jan 24, 2020 at 20:54



As per user(New Developer) request, how to set box detector. You can use like this

use `@MCR` BoxDetector class and then follow these steps.

I am just giving example about text Recognizer so you can set like this

```
TextRecognizer mTextRecognizer = new
TextRecognizer.Builder(getApplicationContext()).build(
BoxDetector boxDetector = new BoxDetector(mTextRecogni
```

set boxDetecotr here

```
boxDetector.setProcessor(new Detector.Processor<TextBl
                @Override
                public void release() {

                }

                @Override
                public void receiveDetections(Detector
detections) {
                    SparseArray<TextBlock> items =
detections.getDetectedItems();
                    StringBuilder stringBuilder = new
                    for (int i = 0; i < items.size();
                        TextBlock item = items.valueAt
                        if (item != null && item.getVa
                            stringBuilder.append(item.
                        }
                    }

                    final String fullText = stringBuil
                    Handler handler = new Handler(Loop
                    handler.post(new Runnable() {
                        public void run() {
                            // here full string(fullTe
```

```
is it scanned.
                        }
                    });

                }
            });
```

Share Improve this answer

Follow

answered May 6, 2019 at 11:20

Dharmbir Singh

**17.5k** ● 5 ● 51 ● 67

1 Hello, I am facing the Exception Detector processor must first be set with setProcessor in order to receive detection results. – Alan Mar 7, 2020 at 14:12