

# Face Detection in Go using OpenCV and MachineBox



Alex Pliutau   Jan 19, 2018

#opencv #machinebox #go

## Face Detection in Go using OpenCV and MachineBox

This is a text version of this video: [packagemain #5: Face Detection in Go using OpenCV and MachineBox](#).

I found a very nice developer-friendly project [MachineBox](#), which provides some machine learning tools inside Docker Container, including face detection, natural language understanding and few more. And it has SDK in Go, so we will build a program which will detect my face. We will also use OpenCV to capture video from Web camera, it also has Go bindings.

MachineBox can be installed very easily by running Docker container. First of all we need to register on [machinebox.io](#) and get key, then we can set environment variable `MB_KEY` and run facebox on port 8080:

```
export MB_KEY="YOUR_MB_KEY"
docker run -d -p 8080:8080 -e "MB_KEY=$MB_KEY" machinebox/facebox
```

Let's go now to [localhost:8080](#) and verify that it's ready, now we can use Go SDK to communicate with facebox from our Go program.

we have to install OpenCV. I am using GoCV.io which has Go bindings for OpenCV. Installation may be complicated, but it worked on my Mac with OpenCV3. Also let's get Go package.

```
brew install opencv3
go get gocv.io/x/gocv
```

Then we need to set some environment variables, we can use `env.sh` from gocv package repo:

```
cd $GOPATH/src/gocv.io/x/gocv
source env.sh
```

Now let's create a `main.go` file and capture video from web camera:

```
package main

import (
    "log"

    "gocv.io/x/gocv"
)

func main() {
    // open webcam. 0 is the default device ID, change it if your device ID is
    webcam, err := gocv.VideoCaptureDevice(0)
    if err != nil {
        log.Fatalf("error opening web cam: %v", err)
    }
    defer webcam.Close()

    // prepare image matrix
    img := gocv.NewMat()
    defer img.Close()
}
```

```

defer window.Close()

for {
    if ok := webcam.Read(img); !ok || img.Empty() {
        log.Print("cannot read webcam")
        continue
    }

    // show the image in the window, and wait 100ms
    window.IMShow(img)
    window.WaitKey(100)
}
}

```

With OpenCV we can also use classifier to recognize faces on the image. Let's do it, for this we'll use Haar Cascades classifier. I already downloaded this XML file to use, you can find it in GoCV repository. With GoCV it's also possible to draw rectangle for each face to highlight it.

```

var (
    blue          = color.RGBA{0, 0, 255, 0}
    faceAlgorithm = "haarcascade_frontalface_default.xml"
)

...

// load classifier to recognize faces
classifier := gocv.NewCascadeClassifier()
classifier.Load(faceAlgorithm)
defer classifier.Close()

for {
    //...

    // detect faces
    rects := classifier.DetectMultiScale(img)
    for _, r := range rects {
        // Save each found face into the file
        imgFace := img.Region(r)
        imgFace.Close()

        // draw rectangle for the face
    }
}

```

```

        gocv.PutText(img, "I don't know you", pt, gocv.FontHersheyPlain, 3, blu
        gocv.Rectangle(img, r, blue, 3)
    }

    //...
}

```

To be able to recognize faces we need to train our model, in Facebox it's very easy, we can just upload few images and set the name. I will do it by saving images of my face into a folder.

```

imgName := fmt.Sprintf("%d.jpg", time.Now().UnixNano())
gocv.IMWrite(imgName, imgFace)

```

Now let's go to [localhost:8080](http://localhost:8080). There are few options to train Facebox: we can use API, we can use web page or we can use Go SDK.

For this example the easiest way is to use form from web page.

Now after we trained it we can check if Facebox can find my face on the image and we can print my name.

```

fbox = facebox.New("http://localhost:8080")

```

GoCV has no option to get `io.Reader` of an image, so we will open our saved file.

```

buf, err := gocv.IMEncode("jpg", imgFace)
if err != nil {

```

```
}

faces, err := fbox.Check(bytes.NewReader(buf))
f.Close()
if err != nil {
    log.Printf("unable to recognize face: %v", err)
}

var caption = "I don't know you"
if len(faces) > 0 {
    caption = fmt.Sprintf("I know you %s", faces[0].Name)
}
```

It's not an advertisement, but I really think that MachineBox is a nice project to provide some Machine learning power to you within simple docker containers.

[Full code of this program on GitHub](#)



**Alex Pliutau** + FOLLOW

I write mostly Go these days. And I'd love to share what I'm doing. If you have something to ask, do not hesitate to send me a message.

[pliutau](#) [plutov](#) [pliutau.com](#)

Add to the discussion



SUBMIT

[code of conduct - report abuse](#)



## Testable Examples in Go

Paweł Słomka - Feb 7



## Go for Java developers (lambda functions to the rescue)

Nicola Apicella - Feb 3



## Concurrent HTTP downloads using Go

Dhanush Gopinath - Feb 5



## Video series: Running Go programs on Raspberry Pi

Alex Pliutau - Jan 31

---

[Home](#) [About](#) [Sustaining Membership](#) [Privacy Policy](#) [Terms of Use](#) [Contact](#)  
[Code of Conduct](#)

The DEV Community copyright 2017 🔥

---