

# Face Detection Web App Tutorial

by Fatih Cagatay Akyon

## Face Detection Flask

Upload an image to see detected faces

Face detection using [Resnet10 SSD Caffe Model](#) | Powered by Python, Flask, OpenCV, Caffe

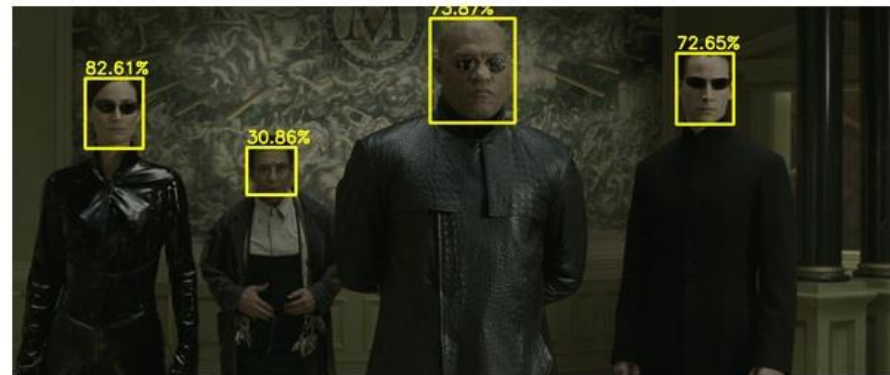
Image size limit: 2 MB

By Fatih C. Akyon - September, 2019

Browse

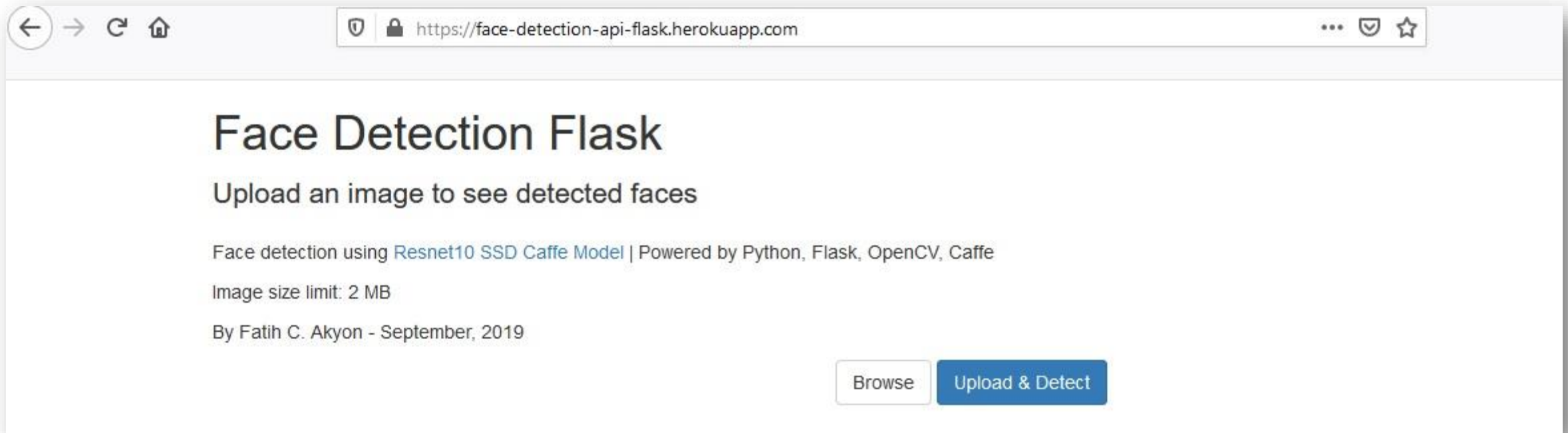
Upload & Detect

Yes! 4 face(s) detected!



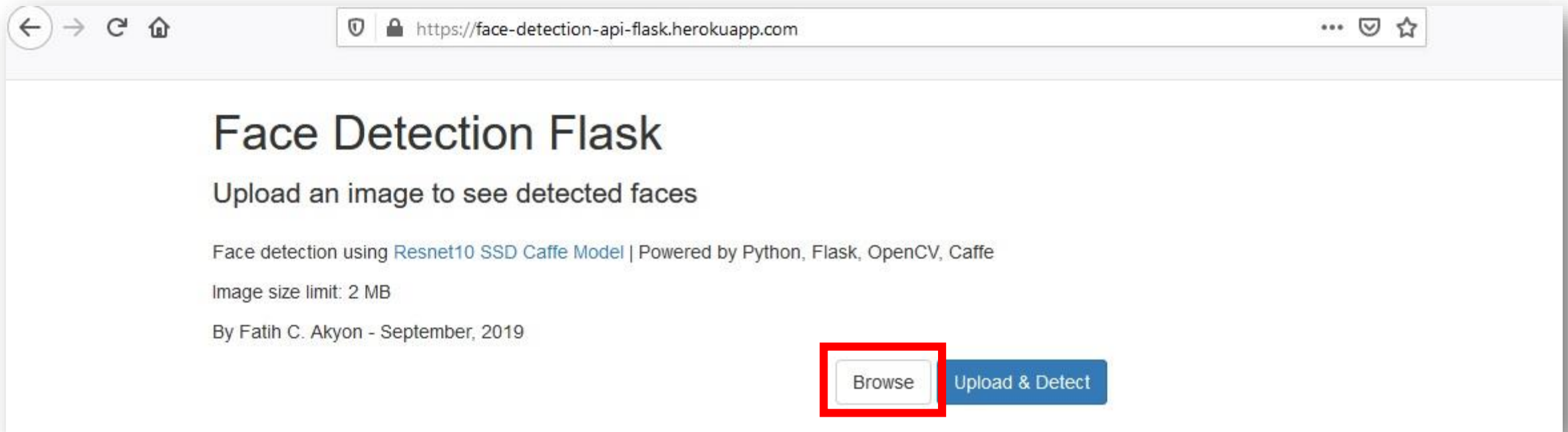
# Web App Preview

- Live demo URL: <https://face-detection-api-flask.herokuapp.com/>



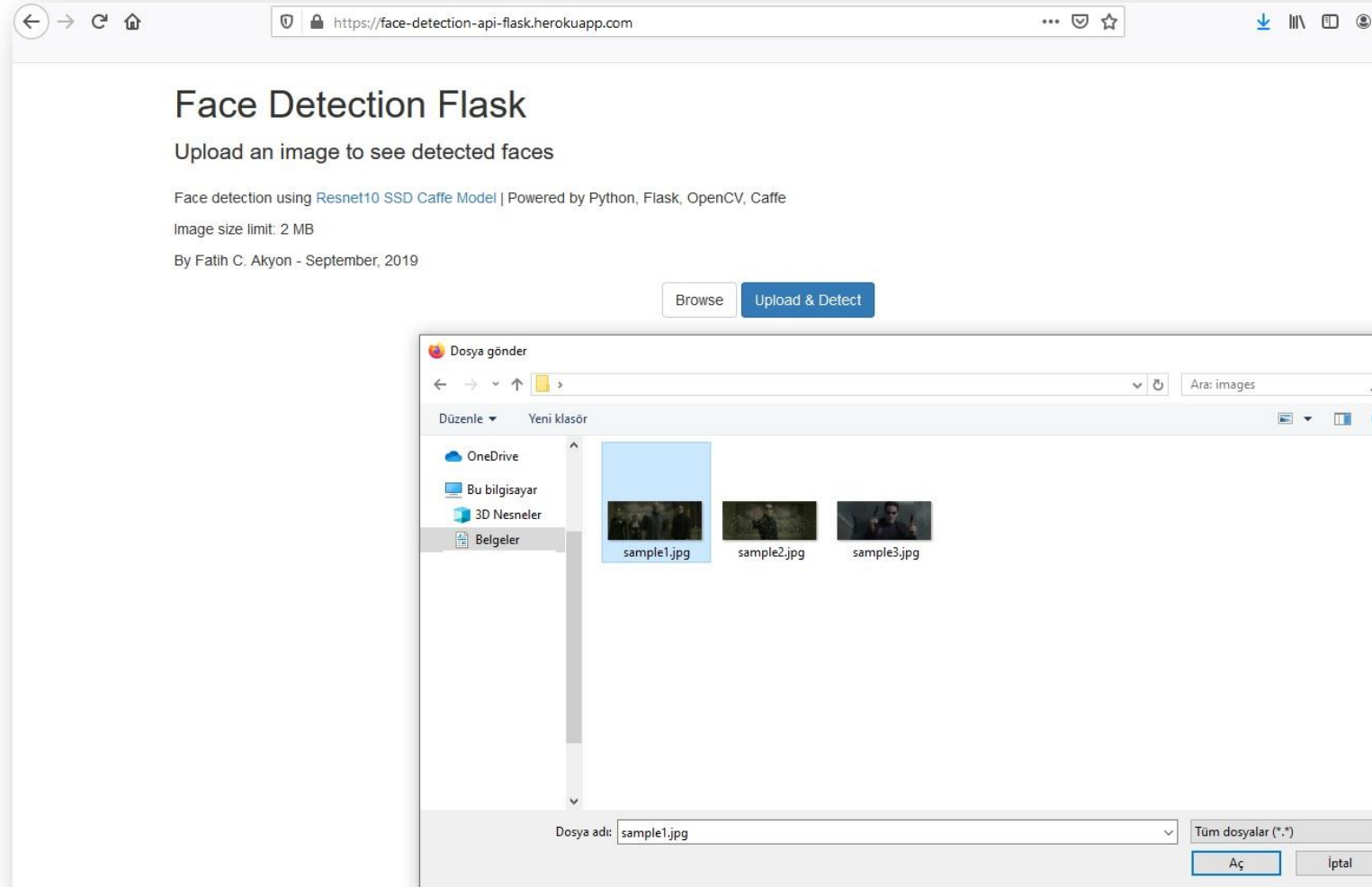
# Web App Preview

- Press “browse” button



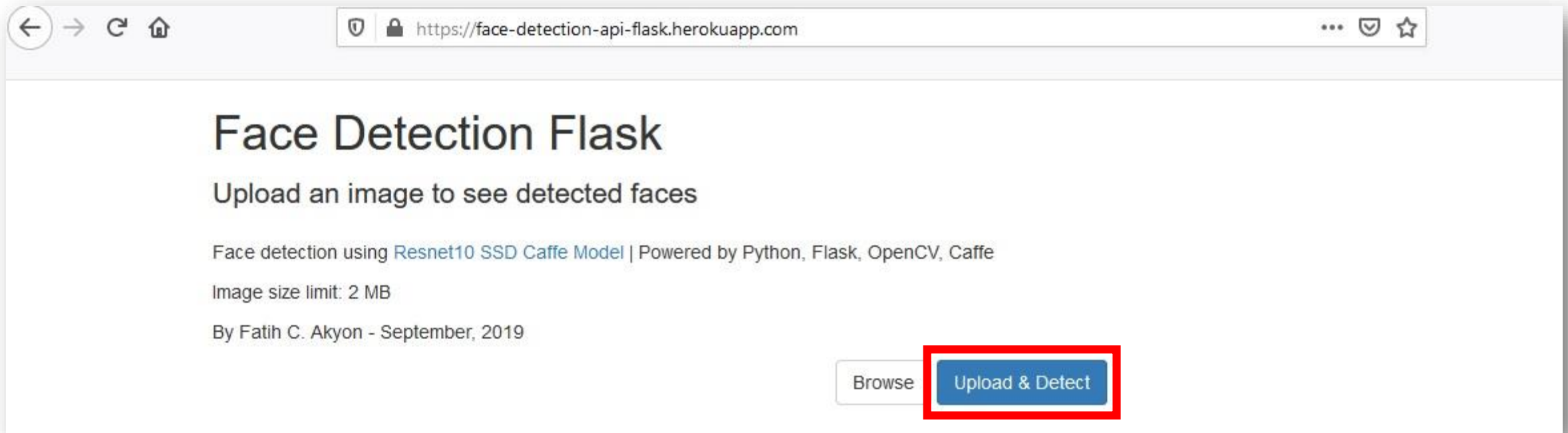
# Web App Preview

- Chose and image file



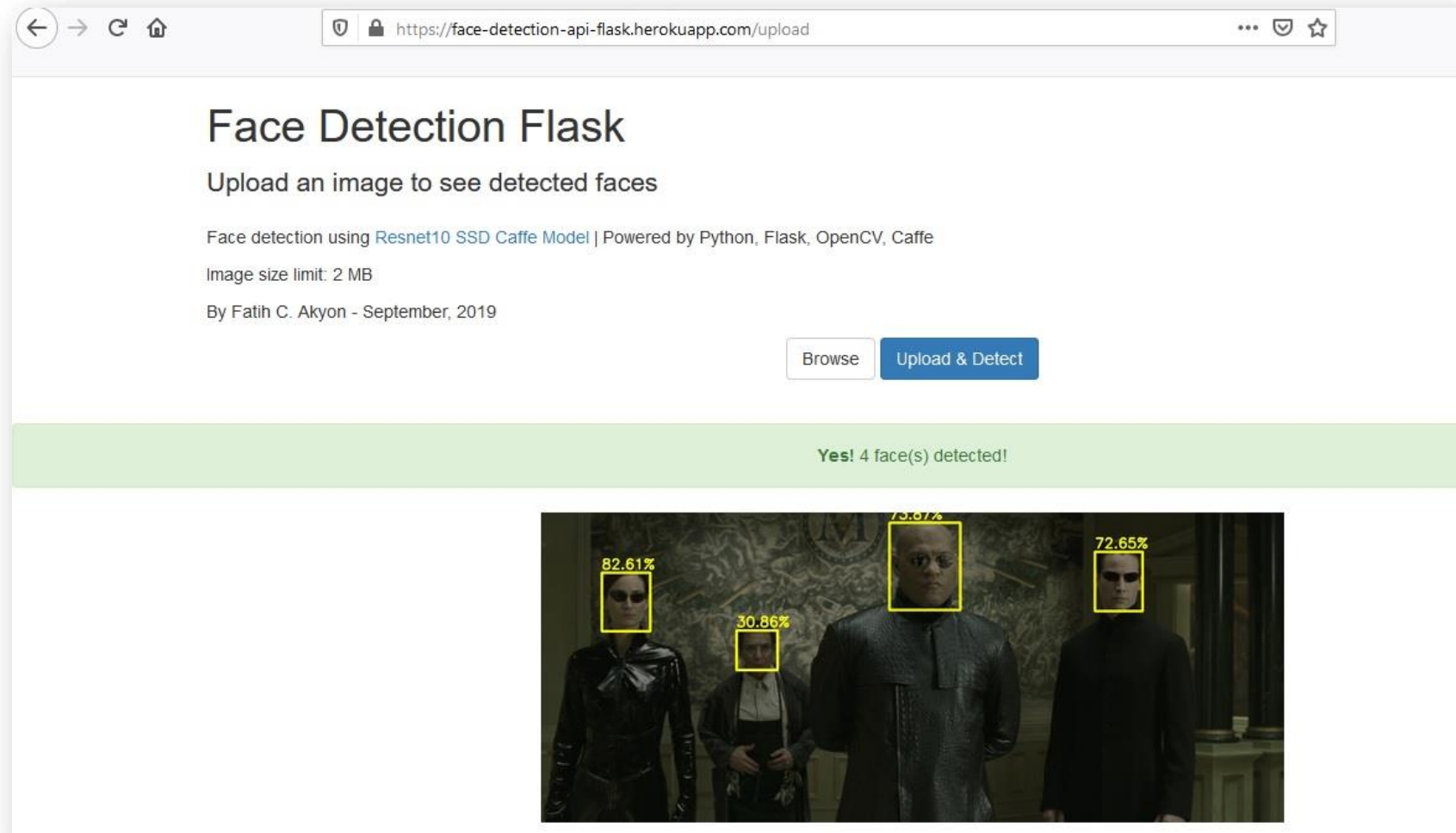
# Web App Preview

- Press “upload and detect” button



# Web App Preview

- Result will show up



# Who Am I

Fatih Cagatay Akyon



**10+** IEEE papers  
**5** patents

Machine Learning  
Deep Learning  
Computer Vision  
Object Detection  
Radar Signal Detection  
Modulation Classification



ORTA DOĞU TEKNİK ÜNİVERSİTESİ  
MIDDLE EAST TECHNICAL UNIVERSITY



**Bilkent University**

**aselsan**



**KUARTIS**



[linkedin.com/fcakyon](https://www.linkedin.com/fcakyon)



[fcakyon@gmail.com](mailto:fcakyon@gmail.com)

# Outline

- Python setup with Miniconda
- Creating our Python environment
- Face detection tutorial
- Creating face detection Flask app
- Git installation and Github account creation
- Heroku account creation
- Pushing our app to Github
- Deploying our app to AWS via Heroku



# Miniconda setup

- Download latest miniconda installer for your platform from:

<https://docs.conda.io/en/latest/miniconda.html>

## Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the `conda install command` to install 720+ additional conda packages from the Anaconda repository.

[See if Miniconda is right for you.](#)

### Windows installers

#### Windows

Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3 Windows 64-bit	51.5 MiB	f18060cc0bb50ae75e4d602b7ce35197c8e31e81288d069b758594f1bb46ab45
	Miniconda3 Windows 32-bit	54.0 MiB	7c30778941d2bba03531ba269e78a108b01fa366530290376e7c3b467f3c66ba
Python 2.7	Miniconda2 Windows 64-bit	50.9 MiB	0647c54058f11842c37854edeff4d20bc1fbdad8b88d9d34d7efda1630e64046
	Miniconda2 Windows 32-bit	48.7 MiB	@d106228d6a4610b599df965dd6d9bb659329a17e3d693e3274b20291a7c6f94

### MacOSX installers

#### MacOSX

Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3 MacOSX 64-bit bash	49.4 MiB	5cf91dde8f6024061c8b9239a1b4c34300238297eddb9ef2061eb9d1a7f69bc
	Miniconda3 MacOSX 64-bit pkg	59.8 MiB	9927f1de5151a1a6431b02046fba089e8b97a55a244f02ffc3207522092907b
Python 2.7	Miniconda2 MacOSX 64-bit bash	39.4 MiB	ed08f4037e40e13eb1d2adc89e054dfb165470cc77be45ef2bf9cb31c8072f39
	Miniconda2 MacOSX 64-bit pkg	47.8 MiB	fcc30b2e18f7a292b34b2e24ad855786a66423f860157fa2b77e40b6392f0ebb

### Linux installers

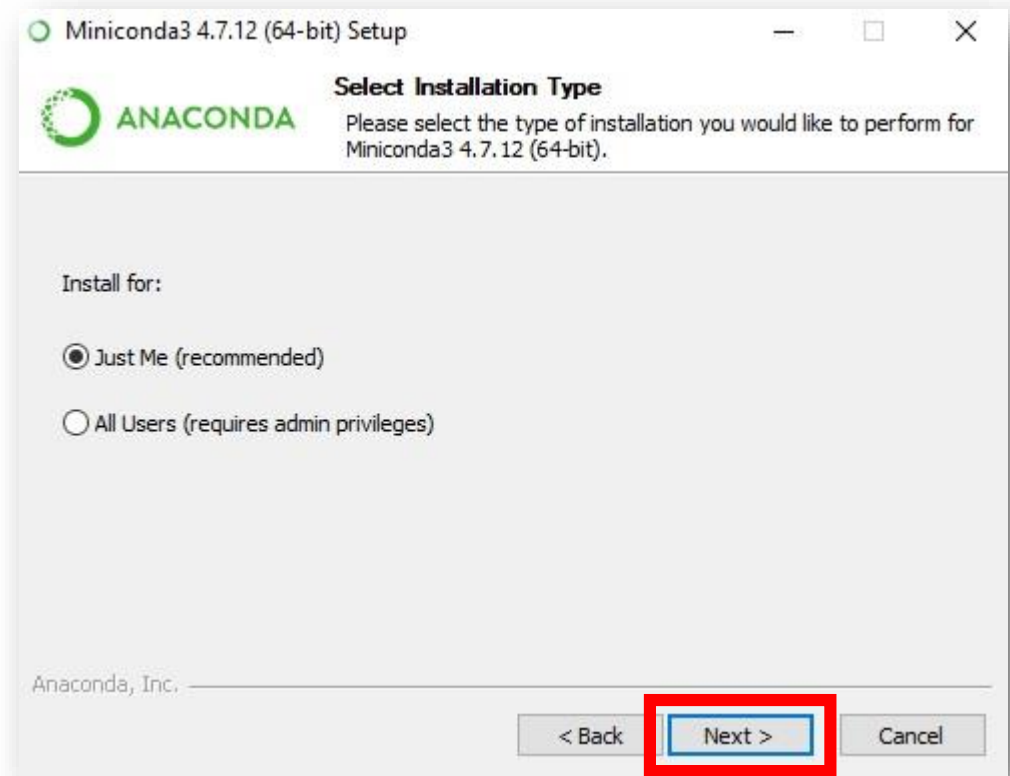
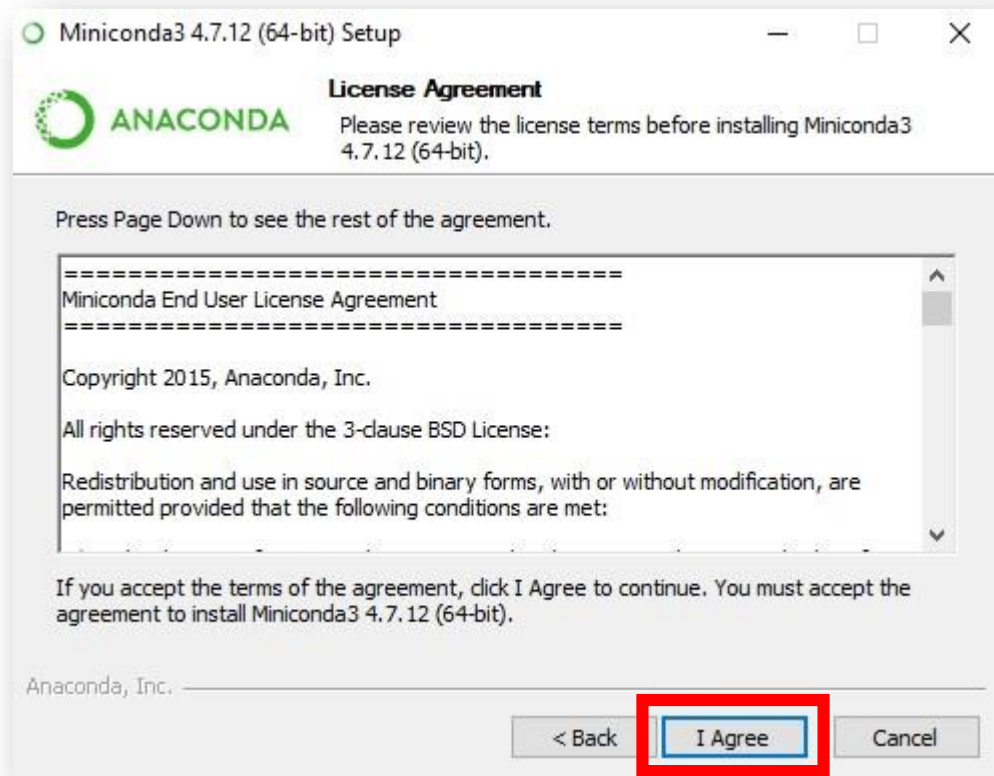
# Miniconda setup

- Download latest miniconda installer for your platform from:

<https://docs.conda.io/en/latest/miniconda.html>

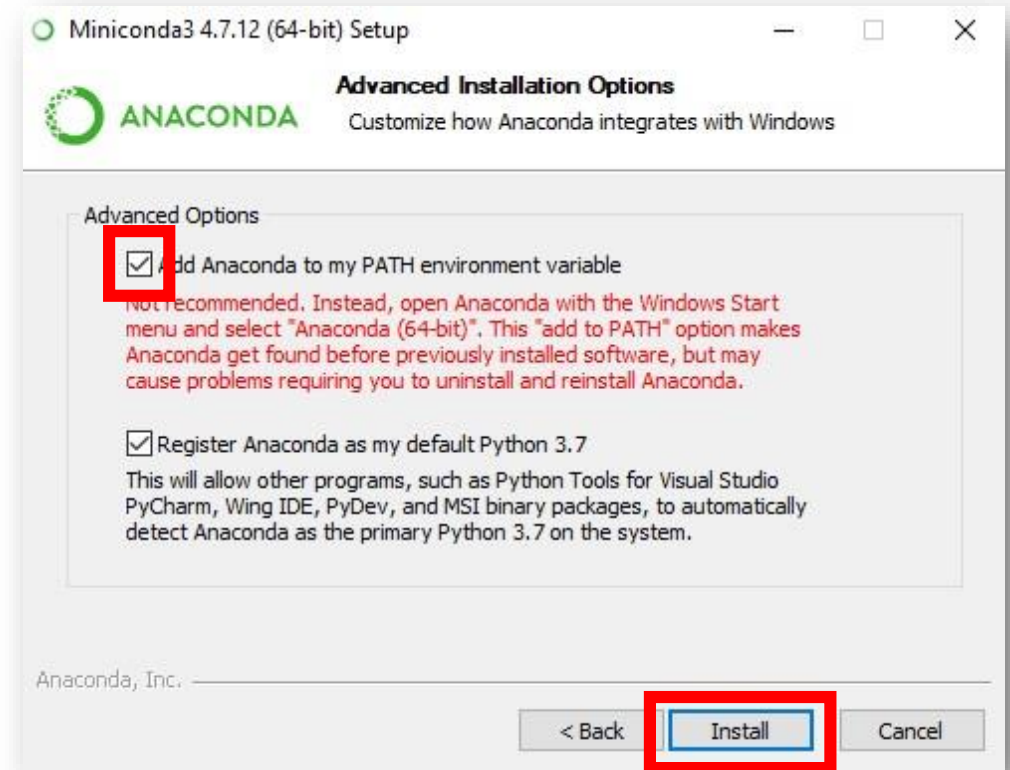
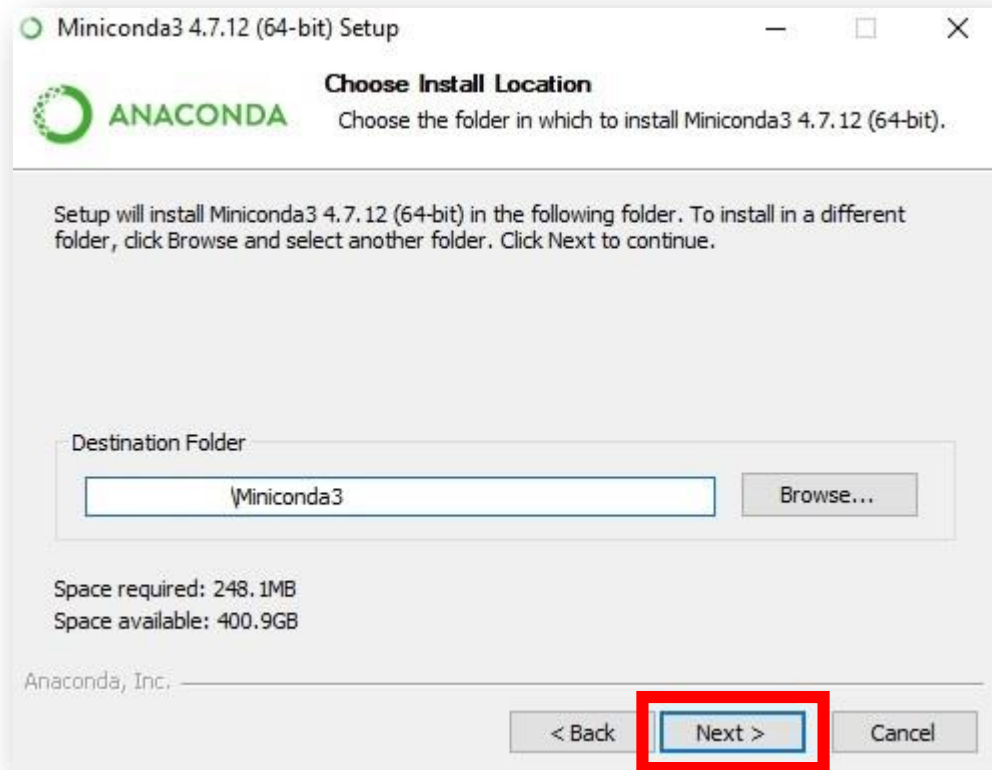


# Miniconda setup



# Miniconda setup

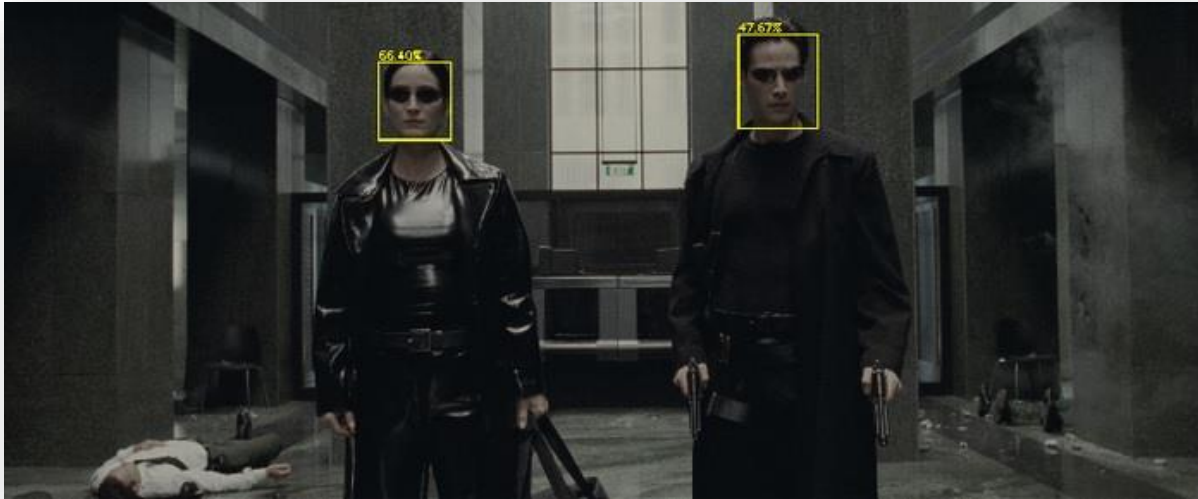
- Don't forget to check the upper box



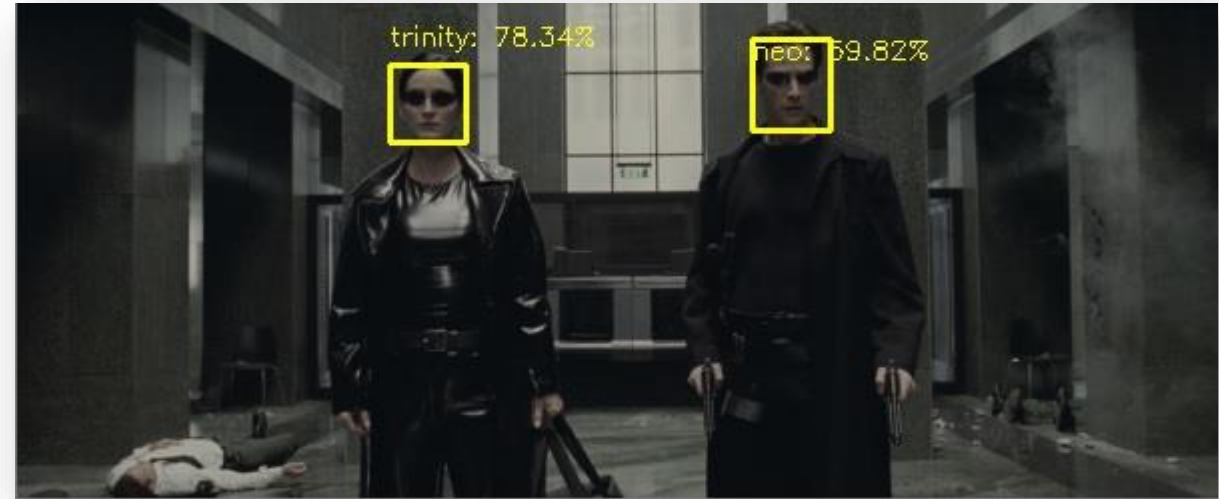
# Python environment setup

- Open terminal (cmd)
- `"cd"` to project folder
- Enter `"conda create --name face-detection-app python=3.6"`  
(To delete enter `"conda remove --name face-detection-app --all"`)
- Enter `"conda activate face-detection-app"`
- Install required packages
- Enter `"conda install spyder"` (For later)

# Face detection tutorial



Face detection



Face recognition

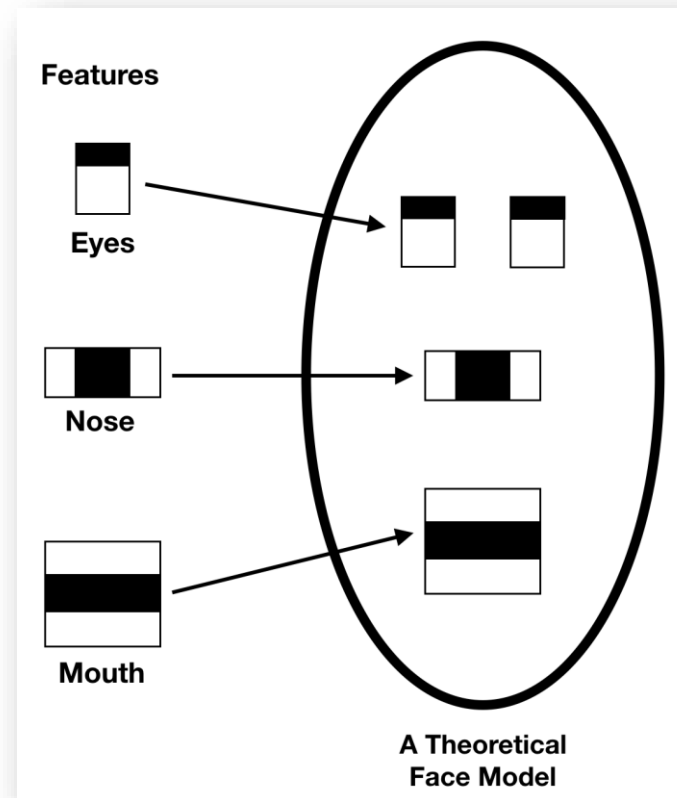
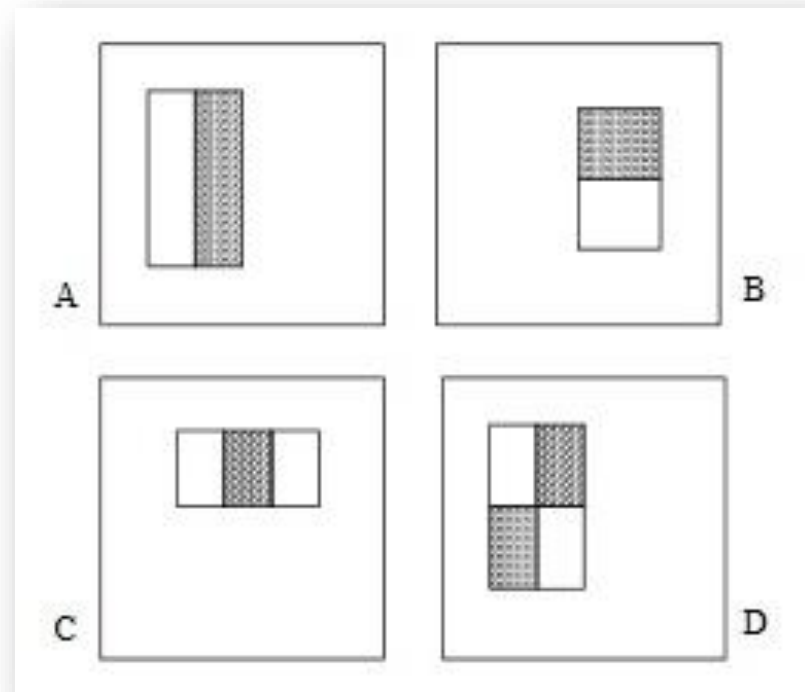
# Face detection tutorial: Facebook example





# Face detection tutorial: Haar cascades

Paper url: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>





# Face detection tutorial: Haar cascades

Paper url: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>

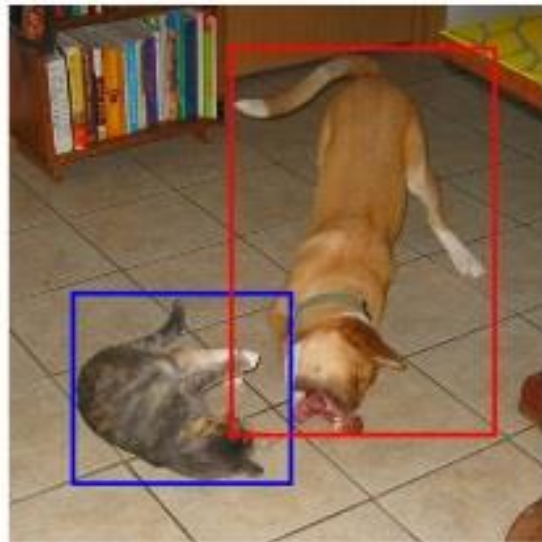
**OpenCV function:** `cv2.CascadeClassifier.detectMultiScale()`

**scaleFactor:** The value indicates how much the image size is reduced at each image scale. A lower value uses a smaller step for downscaling. This allows the algorithm to detect the face. It has a value of  $x/y$ , where  $x$  and  $y$  are arbitrary values you can set.

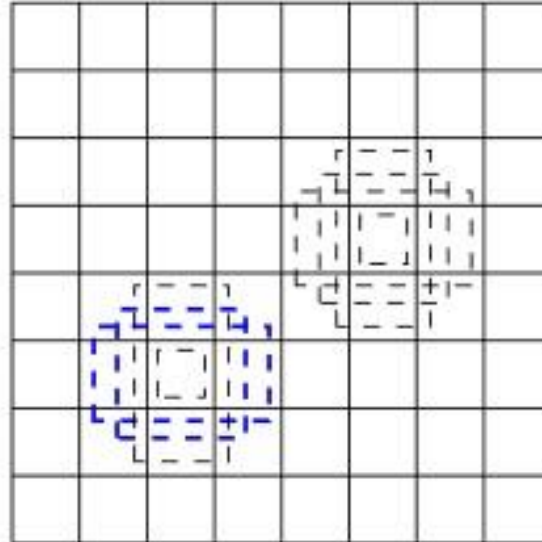
**minNeighbors:** This parameter specifies how many “neighbors” each candidate rectangle should have. A higher value results in less detections but it detects higher quality in an image. You can use a value of  $X$  that specifies a finite number.

# Face detection tutorial: SSD

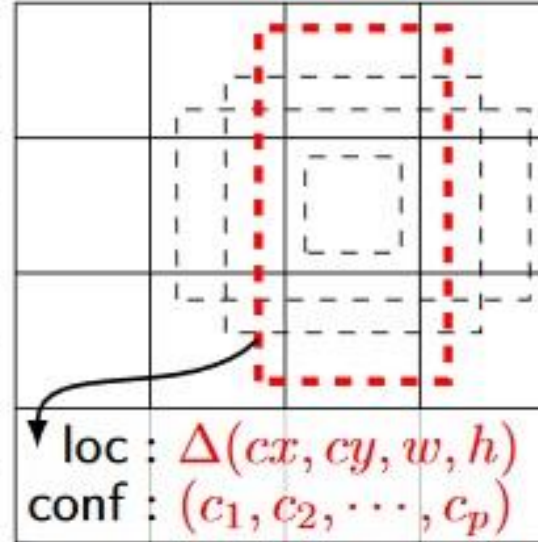
Paper url: <https://arxiv.org/pdf/1512.02325.pdf>



(a) Image with GT boxes



(b)  $8 \times 8$  feature map

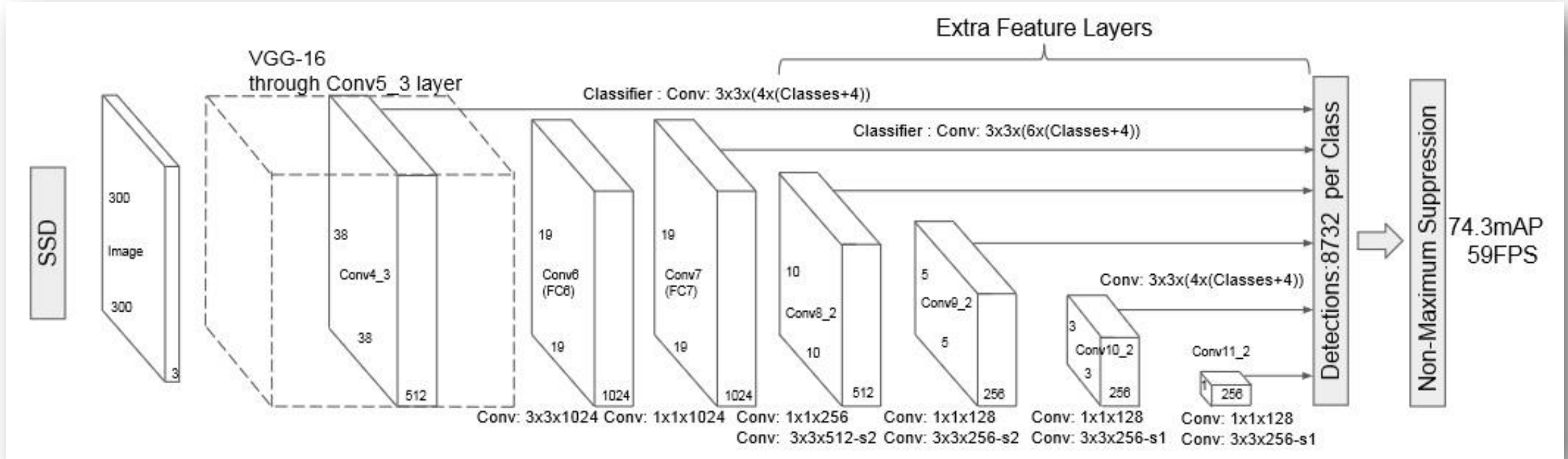


loc :  $\Delta(cx, cy, w, h)$   
conf :  $(c_1, c_2, \dots, c_p)$

(c)  $4 \times 4$  feature map

# Face detection tutorial: SSD

Paper url: <https://arxiv.org/pdf/1512.02325.pdf>



# Face detection tutorial: DL Frameworks



# Face detection tutorial: SSD

Our model = SSD Detector + (WIDER + Fddb Datasets)

`"res10_300x300_ssd_iter_140000.caffemodel"`

## WIDER Dataset

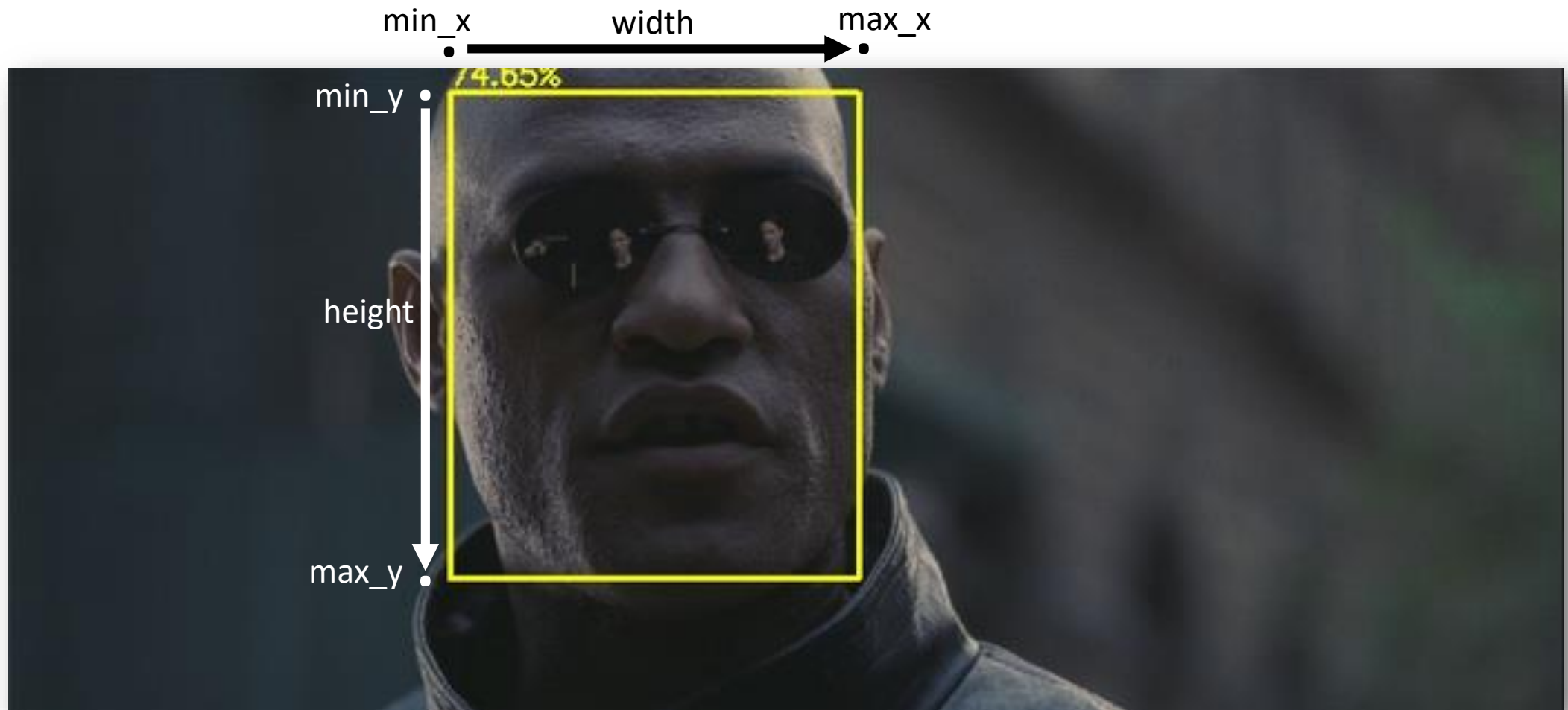


## FDDDB Dataset



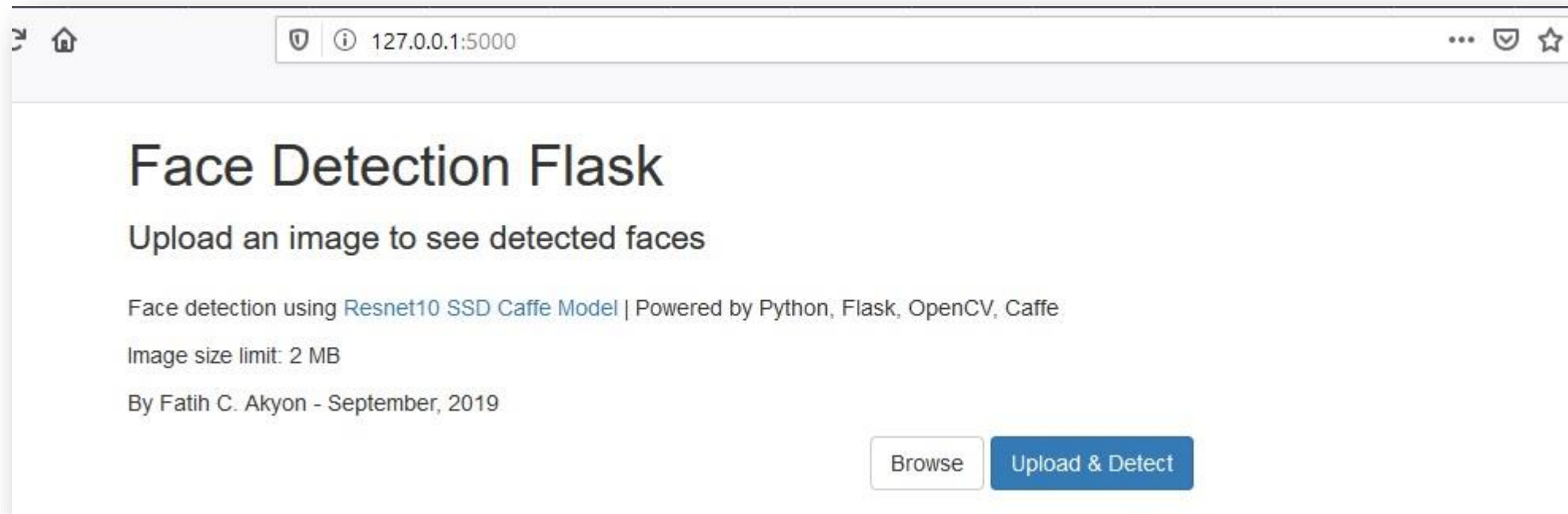


# Face detection tutorial: OpenCV coordinates



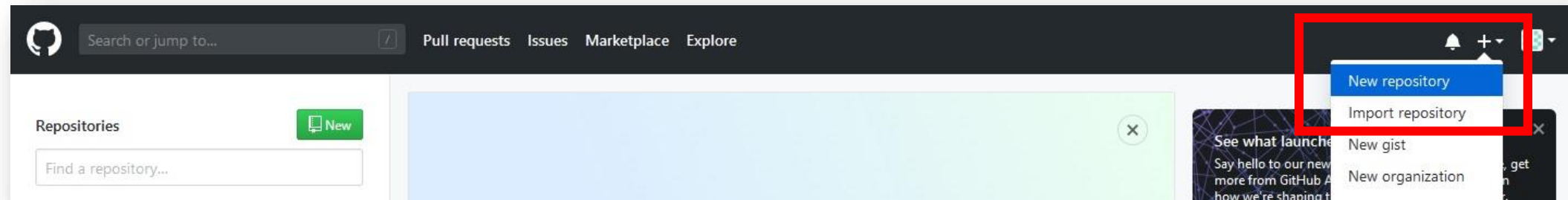
# Face detection tutorial: Local test

- Enter “python app.py” at project directory
- Check <http://127.0.0.1:5000/> on browser



# Git installation and Github account creation

- Download latest git installer for Windows from:  
<https://git-scm.com/download/win>
- Install “*Git-2.24.0.2-64-bit.exe*”
- Create a Github account at:  
<https://github.com/join>
- Create new repository:





# Creating a Github repository

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

### Repository template

Start your repository with a template repository's contents.

No template ▾

---

Owner

Repository name \*

fcakyon ▾

/ face-detection-app ✓

Great repository names are short and memorable. Need inspiration? How about [animated-enigma](#)?

### Description (optional)

A face detection web app powered by SSD face detector using Flask/OpenCV/Heroku

---

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

---

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: Python ▾

Add a license: None ▾ ⓘ

Create repository

Assign your repository name

Descriptive info on project

Good to have a readme file

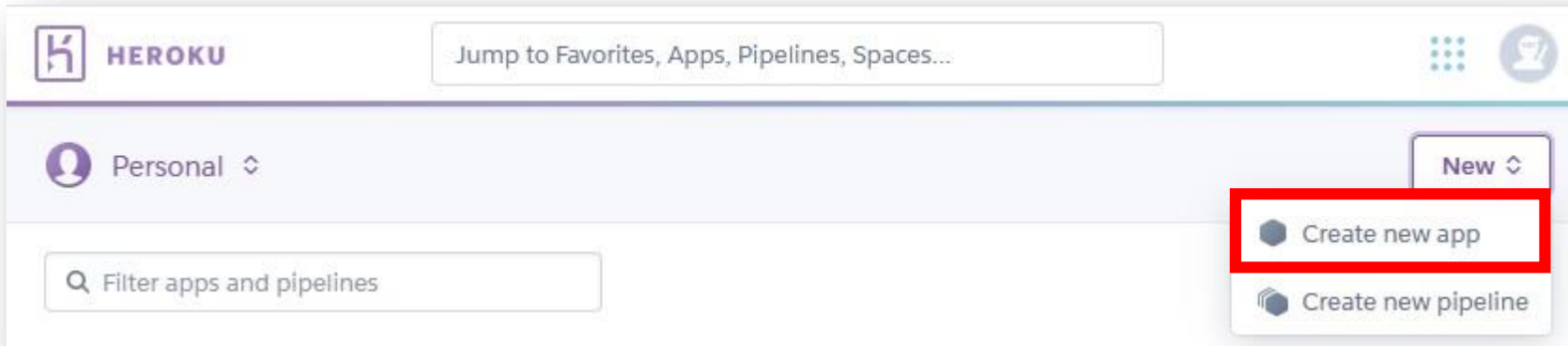
Prevents uploading unnecessary files

# Pushing our project to Github

- Create empty folder
- Open git bash
- Enter "git init"
- Enter "git remote add origin ..."
- Enter "git pull origin master"
- Copy project folder inside this folder
- Enter "git add ."
- Enter "git commit -m 'initial commit' "
- Enter "git push origin master"

# Creating a Heroku Account and App

- Create a Heroku account at:  
<https://signup.heroku.com/>
- Create new Heroku app:
- <https://dashboard.heroku.com/apps>



# Deploying Our App to Heroku

- Connect Heroku app to Github

