Python

1. Artificial Intelligence, Machine Learning, Web Scraping, Rest

* Stock market bot that can predict the future growth of web scraped stocks with a smart intelligent algorithm, plot them, pasist them, and make them accessible on a https python flask server with multiple endpoint resources.
  + bitcoin
* Display plots via. Domain.
* API display information

1. Perception or face recognition

* V1
  + Api that can scan pictures linked to it
  + Persist them to a database
  + Make the data available at an endpoint
* V2
  + Two factor authentication form-based
  + Can go down over a secure connection and match a picture or live camera to a dataset.
  + Something more …
* Find information on people/pictures using reverse image search
* Anti-Face Recognition face painting / masks
* Hi-jack webcam and use with face recog
* Black Hat (Stalker) vs White Hat (Login Authentication)
* Hash/Salt/Peppering

**Disposition(NEW):**

The project will be focusing on the development of a facial recognition service/framework that with a high certainty can securely authenticate users based on webcam footage.

Summary:

We will develop our own facial recognition service/framework that can receive a picture of a person, then through our own neural network match the face with known faces from a database to recognize and authenticate the person in the picture.

1. Facial detect faces in a picture
2. Make a neural network that with a high certainty can recognize faces by matching them with our database of known faces. This part requires a lot of machine learning and training with a large dataset of known faces.
   1. (Stretch Goal) Plot validation loss and validation accuracy during training/testing.
   2. If the face in question is recognized, save the match accuracy to a csv file, and return the correct name of the verified person's face.
   3. If the face is not recognized, run the function again and save the failed match accuracy to a csv file.
   4. (Stretch Goal) Visualize/plot the verification data from the csv.
3. Make a feature that can detect a face and then persist samples of the face in question to a database.
4. (Stretch Goal) If we have time, we could deploy our framework to a droplet running flask so it functions as a cloud service. Here we could also allocate our database.

Concepts and Focus Areas:

The concepts involved in this project in regards to the python course and related technologies include the following entries:

* Data persistence in python
* Logging
* OpenCV
* Working with video capture
* Neural networks, Deep learning
* Machine Learning

Stretch Goals:

* Python web services with flask
* Requests, Headers and Authentication
* Host the neural network training on an external server
* Deployment of framework to a droplet
* GUI
* Login to display 2 factor auth?

**Disposition(OLD):**

The project will be focusing on the development of a facial recognition service/framework that with a high certainty can securely authenticate users based on webcam footage.

Summary:

We will develop our own facial recognition service/framework that can take a picture or a stream of pictures, then through algorithms from our own neural network match the face with known faces from a database to recognize and authenticate the person in the picture.

The application in question will contain a login restrain that can prompt the user for their webcam and stream the data to a droplet running our facial recognition system. This system will authenticate the footage by matching it with the contents of a database droplet where images have been decoded and stored. If the database does not contain a facial scan of the particular user yet, there will be an option to scan the users face through the client application, send it to the service droplet, which can then persist the packages to the database droplet so that the user for future purposes can login using face recognition.

Practical demo:

The system will be developed in python as a demo concept, demonstrating the usage of a face recognition framework that we ourselves are going to build, as well as some of the python concepts and methodology that we have learned throughout the semester. This system will also be part of a crossover project between the courses Python and Security.  
  
Concepts and Focus Areas:

The concepts involved in this project in regards to the python course and related technologies include the following entries:

* ~~The python face recognition library~~
* Python web services with flask
* Data persistence in python
* Logging
* Requests, Headers and Authentication
* ~~Working with modules~~
* OpenCV
* Working with video capture
* Machine Learning
  + Calculation of tolerance
  + How to recognize a face
    - Is there one face or more?

**Kilder:**

Security and biometrics (feb 2020):

<https://books.google.dk/books?id=vCbTDwAAQBAJ&pg=PA154&lpg=PA154&dq=owasp+face+recognition&source=bl&ots=FuqtKRL6_O&sig=ACfU3U3q-VQMVnncc0scuZg_GtmBzG8-9g&hl=da&sa=X&ved=2ahUKEwjI97-UrojpAhUqx4UKHTRMAHsQ6AEwCHoECAoQAQ#v=snippet&q=owasp%20face%20recognition&f=false>

How facial recognition algorithms can be tested for algorithmic bias (jan 2020):

<https://ictinstitute.nl/face-recognition-algorithms-algorithmic-bias/>

The vulnerability of face recognition systems to spoofing mask attacks (oct 2013):

<https://ieeexplore.ieee.org/document/6638076>

Vulnerability of Face Recognition Systems Towards Morphed Face Attacks (2017):

<https://christoph-busch.de/files/Scherhag-VulnerabilityFaceMorphing-IWBF-2017.pdf>

Security challenges in the age of facial recognition (nov 2019):

<https://securityboulevard.com/2019/11/face-off-security-challenges-in-the-age-of-facial-recognition/>