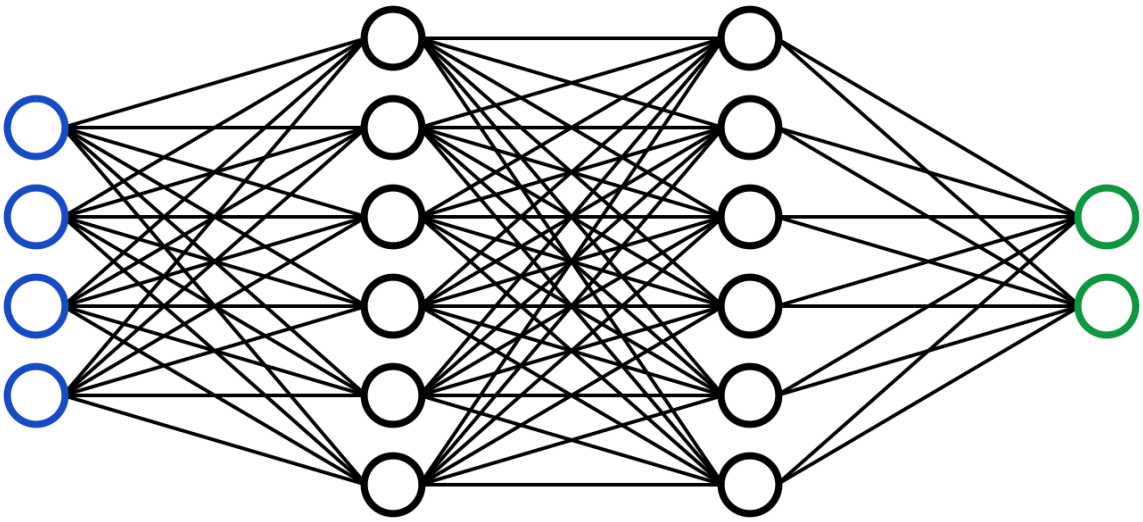
Facial Recognition Software



Jorge Zepeda(j_zepeda3@u.pacfic.edu)

Cooper Riley(c_riley9@u.pacific.edu)

09.06.2021

SYSTEM ARCHITECTURE

Software Modules:

The programming language that we will be using will be Python. We will also be using opencv which is a free computer vision software library for machine learning. We will use the flask web framework to get our program running on a webpage. We will be using the user's imputed image to train our model.

Hardware Components:

Our hardware components will be limited as this will be run on a webpage. The hardware requirements will be any computer with network connectivity.

User Interface:

The user interface will be kept simple. The user will interact with one page. The user will be prompted to drag and drop a video that they find suspicious. Once the user clicks upload they will be greeted with the percentage out of 100% on how confident the model thinks is the right face.

Interfaces to external systems:

An AWS account will be needed for us to host our website. In addition we will use the Anaconda distribution to simplify the package management and deployment.

HARDWARE, SOFTWARE, AND SYSTEM REQUIREMENTS

Hardware requirements:

Minimum Requirements:

- **Processor**: 1.9 GHz
- **Memory**: 2 -GB RAM

Network:

- The user will need an internet connection. They will need to be connected either through wifi or ethernet.

Software Requirements:

- Python 3.6 - 3.9
- TensorFlow 2
- AWS

System Requirements:

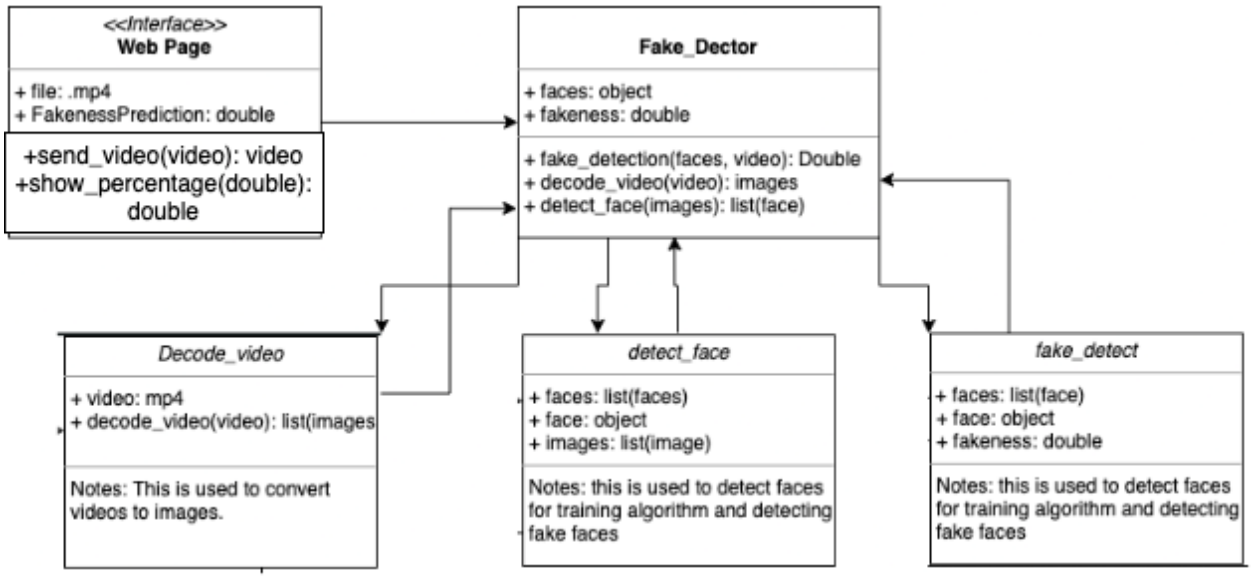
- **OS**: Windows 7, 8, 10, Mac, Linux

EXTERNAL INTERFACES

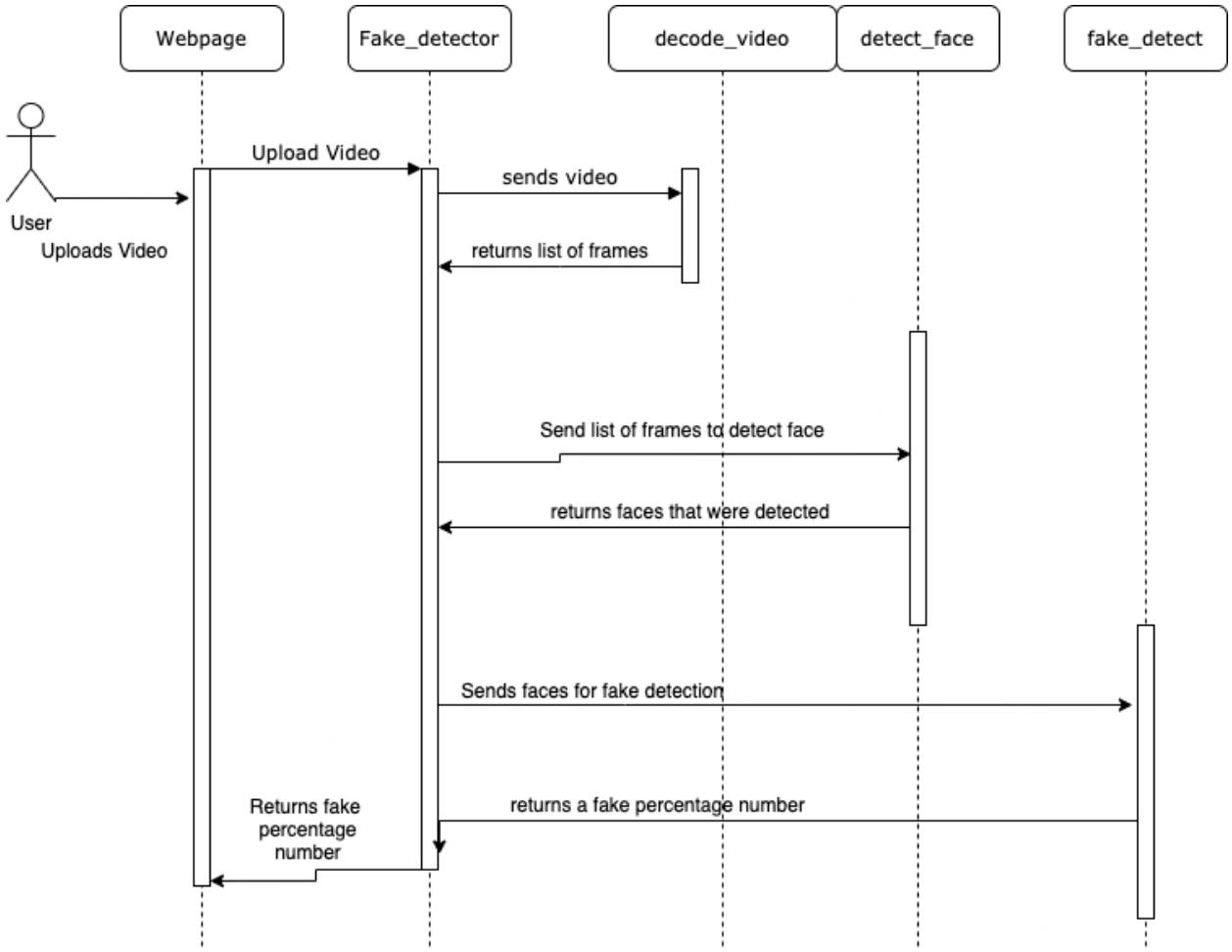
1. AWS
    a. A AWS account will be needed to retrieve the facebook data set. We will need to set up an IAM user and access key set up. We then use

SOFTWARE DESIGN

UML Diagram:



Sequence Diagrams:



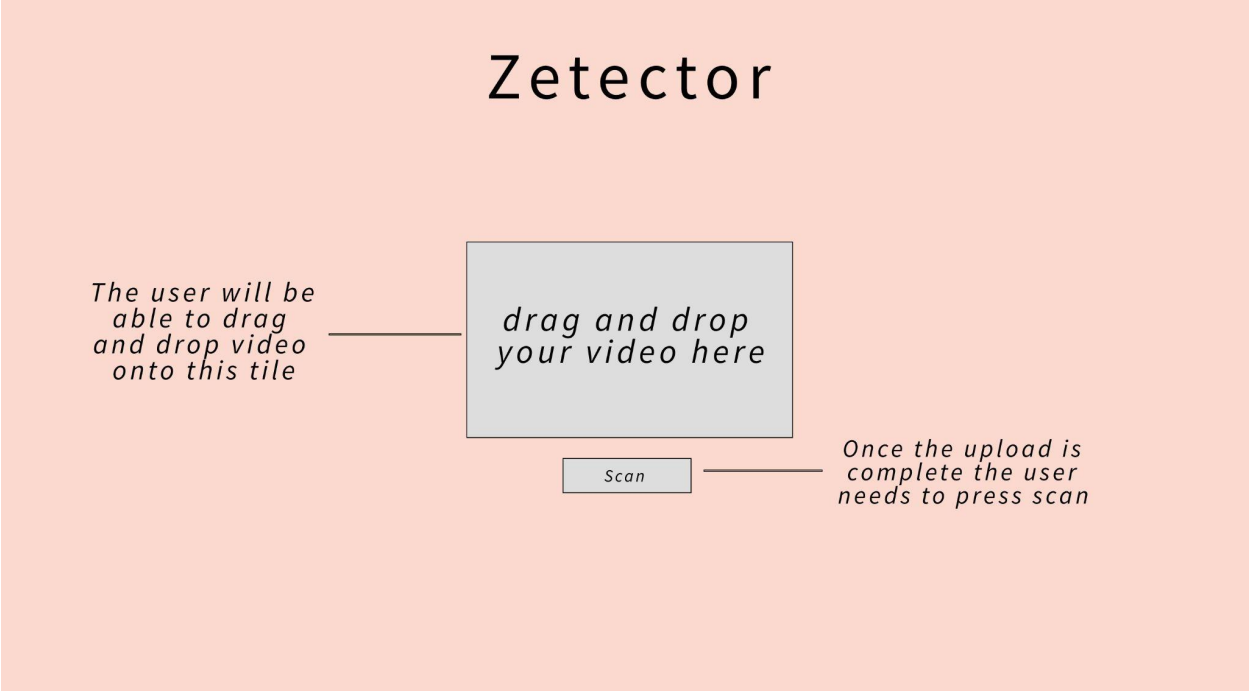
Design Considerations:

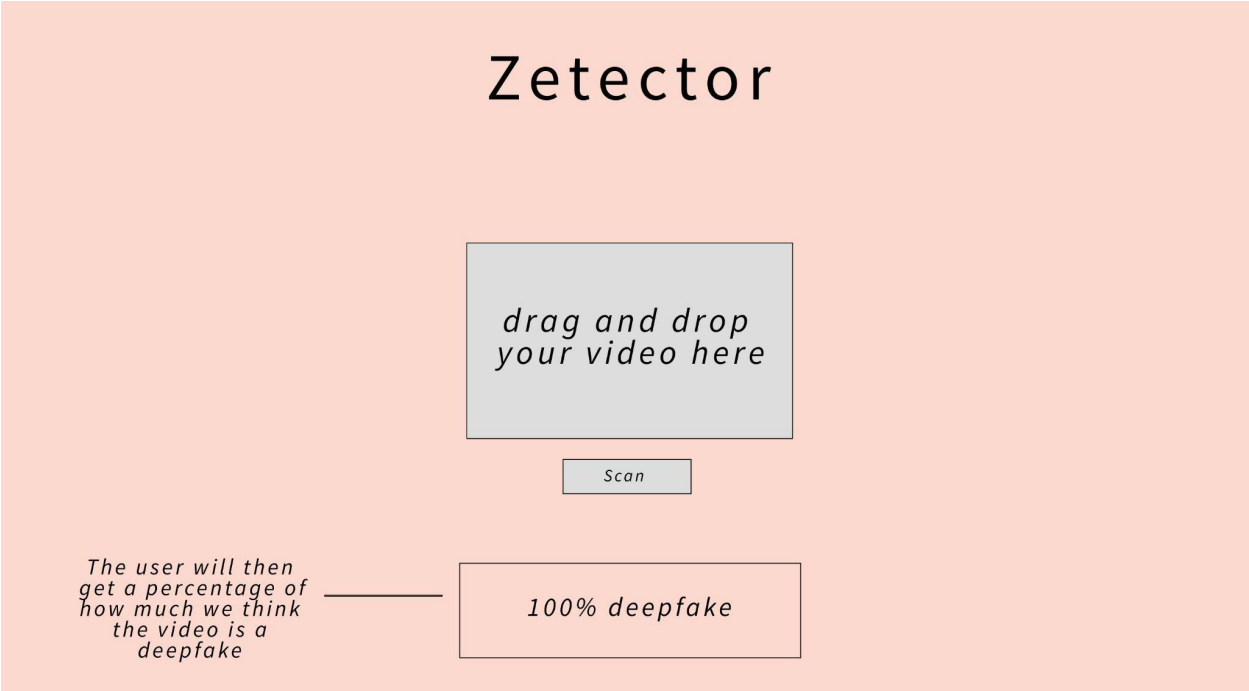
- The program begins on our web page. This is where the user will first drag and drop a mp4 video to get passed into our detection software.
- One the user inputs the video the Decode_video class will take the video as a parameter and begin to parse through the video and detect the number of frames, and return a list of images.
- Detect_face class will take the list of images and detect the faces that are in the video.

- We will then cluster the faces to be able to detect what a face is since there might be too much noise in the dataset videos.
- Once everything has been prepared the faces are ready to go into our classifier to get a confidence level.

USER INTERFACE DESIGN

The user interface will consist of one page. That will change dynamically when the user drags a video they find suspicious into the browser.





GLOSSARY OF TERMS

AWS - Amazon web services

DFDC - Deep Fake Detection Challenge

SSMS - SQL Server Management Studio

REFERENCES

Deepfake detection challenge dataset. Facebook AI. (n.d.). Retrieved September 13, 2021, from https://ai.facebook.com/datasets/dfdc/.

Image classification : Tensorflow core. TensorFlow. (n.d.). Retrieved September 13, 2021, from https://www.tensorflow.org/tutorials/images/classification.

Schleifer, S., Serrats, M., & Soley, M. C. (2010). *Cloud9: Rooftop architecture*. Amazon. Retrieved September 13, 2021, from https://docs.aws.amazon.com/cloud9/latest/user-guide/sample-python.html.