Weekly meeting reports

Date: 11

Time: 1:30 - 2:30

Who showed up: Luca deVerteuil , Stanley, Hong

## Expectations:

### Goal of meeting:

Goal of this meeting is to show the client what we have accomplished so far. More clarification on the project. As well as the milestone we hope to accomplish and by when

## Functional requirements:

* Web application for classification task that presents classified and to be classified images.
* Active learning algorithm implementation, following the principles of clean code.
* Source code integrated in a private source code management system (e.g. GitHub, Gitlab). Including configuration and installation documentation.
* Data storage system for vectors (codes) that are used by the active learning algorithm and a storage system for the images.
* Workflow that encodes the images from a given generated water pipeline inspection and stores the codes and images.
* Performance tests for the time needed for one step of the active learning process.
* Software architecture and technical documentation.
* Enhanced water pipeline inspection generator (optional).
* Classification performance for self-defined classification tasks (optional).
* all code should be written in Python
* all Python code must be formatted with black
* fully test components with flake8, pylint and pytest
* 100% test coverage for components
* use standard tools for all components (logging, argparse)
* limit dependencies
* use numpydoc for docstrings, use type hints for documenting types

| Milestone | Functional Requirement |
| --- | --- |
| Milestone 1, (21/10) | N/A |
| Milestone 2, (02/12) | Have a Working GUI with and make sure the GUI meets the expectation of Rosen client   * Should be able to upload and view photos on website * Should be able to login to website * Start training machine learning |
| Milestone 3, Feburary | By this milestone we should have a working machine learning as well as our own tested test photos from blender working  As well as staring fundamentals and research into vector database |
| Milestone 4, April | By this point the product should be working. Maybe a minor amount of testing needs to be done |

## **Tech stack**

| **Front-end** | **Pros** | **Cons** |
| --- | --- | --- |
| Dash (Preferred) | •Perfect for Dashboarding and data visualization  •Good callback debugger  •Highly scalable | •Does not really have any disadvantages as dash is written for data scientists |
| React | •Good for Dynamic web pages  •Components are reusable  •Ease of testing | •React evolves quickly  •Lacks proper documentation  •Lacking in features for a framework |
| Vue | •Lightweight with reliable performance  •Good documentation  •Supports two-way bindings | •Lack of libraries compared to other frameworks  •Difficulty in scaling |

| **Back-end** | **Pros** | **Cons** |
| --- | --- | --- |
| Django | •Can be used for both front-end and back-end  •Built with scalability in mind  •Ready to use and customizable admin features | •It is not ideal for small projects with minimal features  •Steep learning curve  •Unable to handle multiple requests simultaneously |
| ExpressJS | •Capable of handling multiple requests  •Customizable  •Quick deployment | •Not ideal for CPU intensive work  •Lacks standardization  •Lacks support for asynchronous errors |
| Flask | •light-weight framework  •Suitable for small projects  •Customizable | •Does not have built-in support for basic database system  •Slow deployment of minimum viable product  •Limited community support |

## 

## 

## **Machine Learning algorithms**

| **Algorithms** | **Pros** | **Cons** |
| --- | --- | --- |
| Google AutoML | •Easy to use | •Model train time caps at 1 hour for free users |
| Keras(VGG16) | •High accuracy in image classfication | •Requires a lot of training data |
| Keras(Resnet50) | •Faster and more accurate than VGG16 | •It has deeper network so requires longer training time  •Requires a lot of training data |
| SKLearn(Support Vector Machine) | •Works well with high dimensional data | •Does not perform well when there is a lot of noise in dataset |
| Self-written | •Gain knowledge in machine learning  •Specific to client's need | •Time consuming  •Might not be as efficient  •Outside of skill set |

## Vector data-base:

We are thinking of using weaviate as a vector database as it is an api that is written in python. Its goal is to make a vector database easy as it is a challenging task and the goal of weaviate is to make it alot more available to everyday data scientists.

## 

## Moving forward(Y frames):

Moving forward in these next couple of meetings to be able to gain a better grasp of what works for you we would like to make a storyboard of the product. We will also start gaining familiarity with dash framework(HONG, Stanley) and blender(LUCA) next week.

## Meeting notes

No difference between admin and users

Should support video

Save the progress of what was happening before

Amex sealing (add to classification)

Time stamp of where to fish was found

Just apply pretrained vgg 16 network:

1 active learning:

SVM

Psychic learn python supports svm