System Description

- *Purpose of System*

  Given a 2D array representing a patient brain scan, this system will classify areas of the brain that are at high risk for syndromes associated with misplaced lesions.

- *Basic overview of functionality of the system*

  The user will be prompted to upload a CSV file with particular formatting and select which syndrome they are searching for. The CSV file is then parsed and converted into a 2D matrix of ones and zeros. The matrix is run through the Support Vector Machine algorithm which returns weights specified to each voxel (a pixel in the array). The weights are applied to each voxel, highlighting the areas of high risk. The application then verifies if the area classified as high risk is in the specified region based upon the user-specified syndrome and visualizes the results.

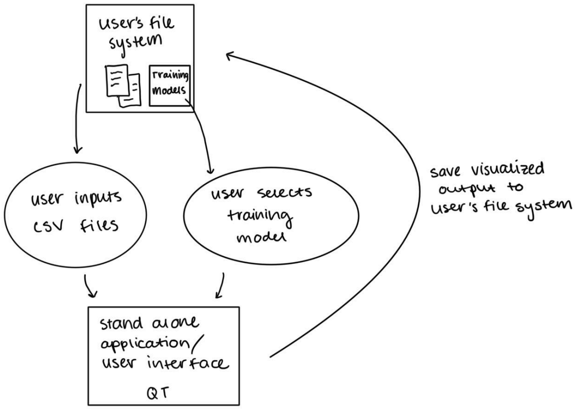
Architectural Review

- *Describe design/architecture of system*

  The system first processes the data to ensure that it is in the correct format. After ensuring correct data input, it is inserted into a machine learning model for parameter optimization and lesion behavior analysis. The results from the machine learning algorithm are then inserted into an image visualization tool that provides various details on the results.

- *List of major system components*
  - Data Preprocessing Tool
  - Image Processor
  - Lesion Localization Model
  - Image Visualization Tool

- *System diagram*


user's file system
Training models
user inputs csv files
user selects training model
stand alone application/ user interface
QT
save visualized output to user's file system

Four Epics

1. As a user of MediViz,
   I want a visualization of the program output
   so that I can analyze areas of high risk and lesion placement.

2. As a user of MediViz,
   I want to store and access my previous data
   so that I can decrease future computation time.

3. As a user of MediViz,
   I want a customizable interface (parameters, targeted region, etc)
   so that I have more control over the algorithm's target

4. As a user of MediViz,
   I want to have the option to utilize systematic occlusion
   so that I can isolate particular voxels for the algorithm to focus on/ignore.

Non-functional Requirements

- The machine learning algorithm should produce output in 10 minutes or less.
- The application should take in files of at size at most 1 megabyte
- Double click to open

Technologies and Frameworks

- Python
- SkiLearn (SVM)
- Pandas
- PlotLib
- Postgres DB

Minimum Viable Product (MVP)

- Basic user interface/upload point for user data
- Machine learning algorithm that provides weights
- Output of voxel weights as a matrix

Preliminary Roadmap

Sprint 0 Feb 6th): Solidify planning, research packages needed
Sprint 1 (Feb 20th): Solidify brain behavior data format & generate data
Sprint 2 (Mar. 6th): Test & finalize app development program
Sprint 3 (Mar. 20th): Implement front end components & running of machine learning algorithm
Sprint 4 (April 3rd): Implement results visualization dashboard + results export

Sprint 5 (April 15th): Implement systematic occlusion
Sprint 6 (April 29th): Review finalized product with Dr.McAfee