Face Mask Detection Project

CS 5500

Team Members

Python Development Team

Zihao Qiu

Yuqi Tao

Robert Dragomir

Aushee Khamesra

Web Development Team

Shruthi Raghuraman

Julia Rakas

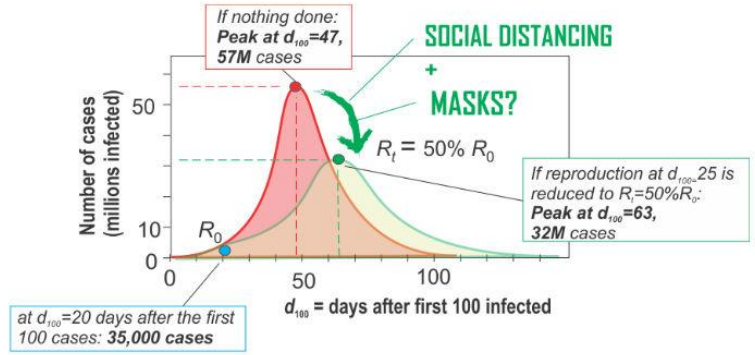
Jingyang Zheng

Randy Lirano

Purpose of the Application

- COVID-19 has a high infection rate which needs to be controlled
- To flatten the curve, masks are recommended
- Enforcing the use of masks in communities is a challenge

- **Solution:** Automate the monitoring
  - Inspired by current automatic temperature measurement applications
- Tool follows mask-wearing CDC protocols
- Tackles the problem in the time feasible time frame
- Increasing the effectiveness of enforcing mask use in public places



[Image Source](#)
[CDC mask source 1](#)
[CDC mask source 2](#)

Major Feature

Python

- Real-time facemask detection
- Deployed on web, or any machine that can run Python

Web

- Test model with image/live stream on device camera
- Easy to connect with team
- Application hosted on Heroku server as part of immediate future work

Data Metrics

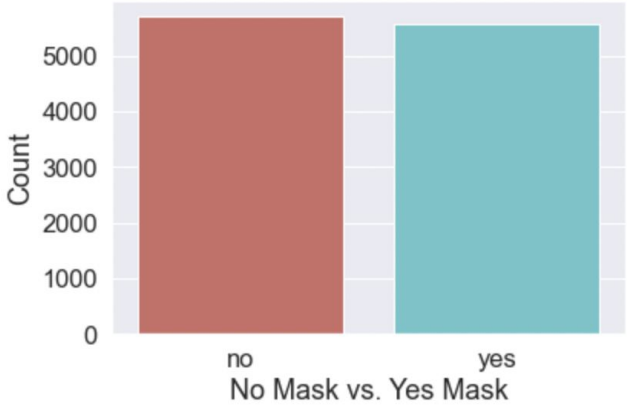
The dataset contains 11,264 images, where:

- Number of images with NO mask : 5701
- Number of images with mask : 5563

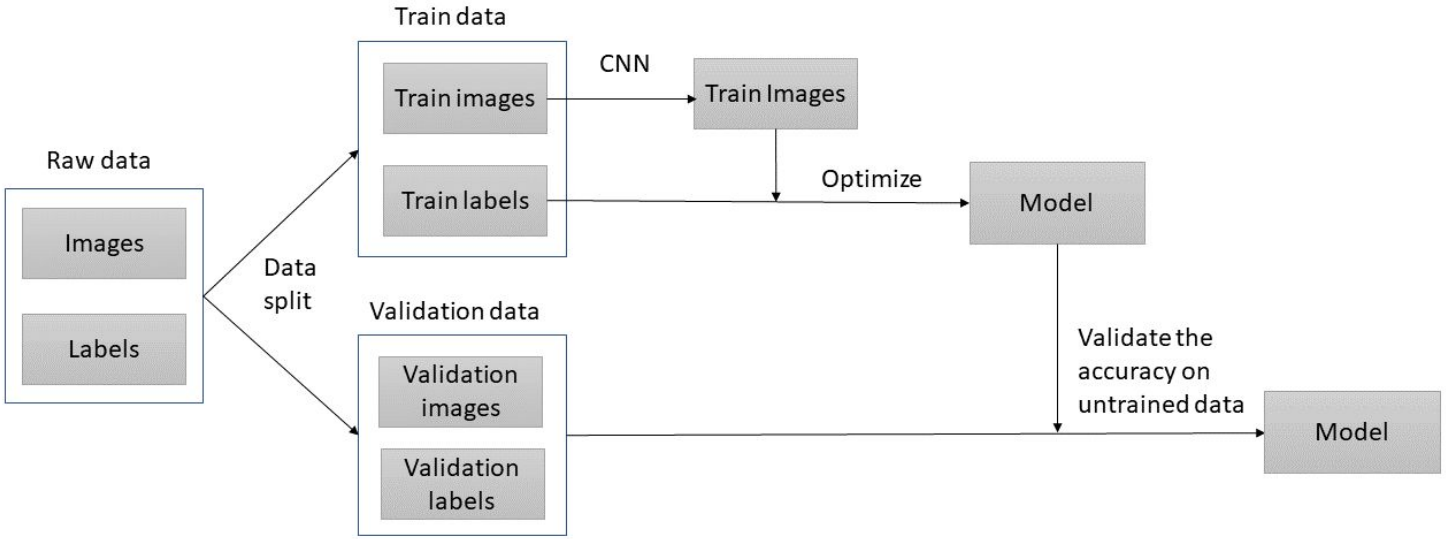
Statistics of the dataset

	label (0/1)
count	11264.000000
mean	0.493874
std	0.499985
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

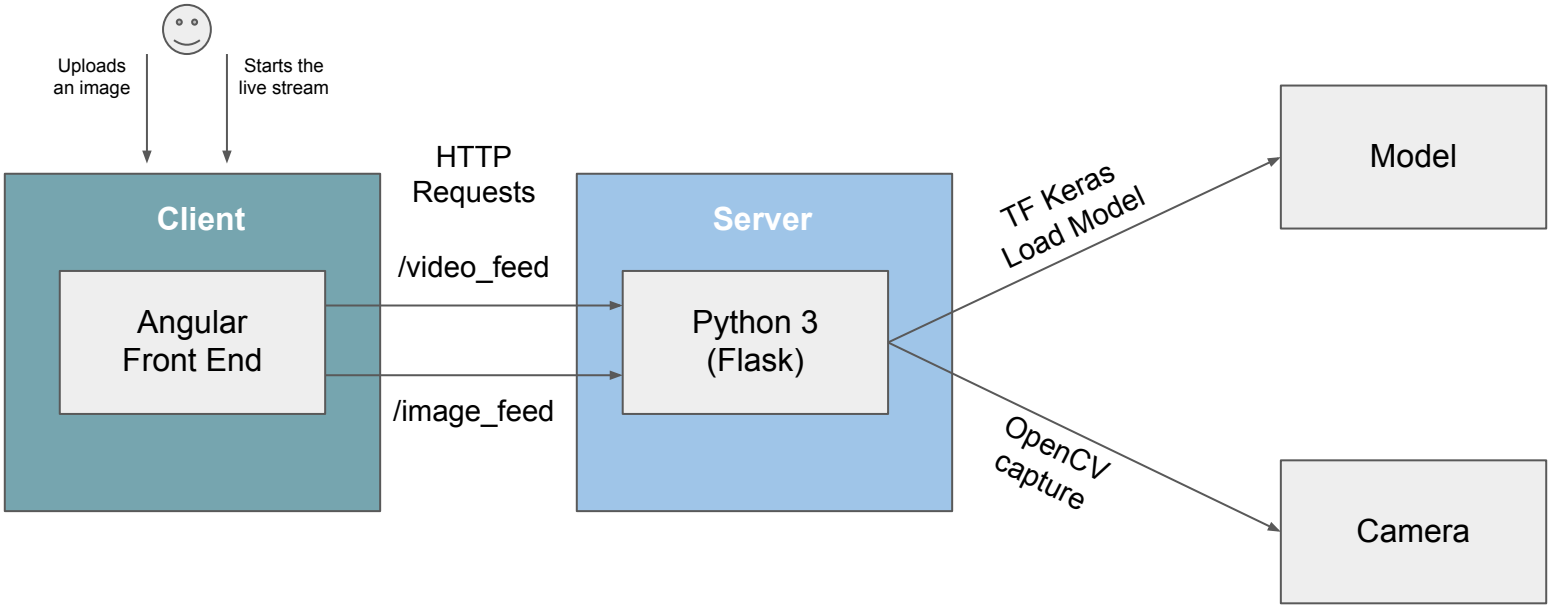
Data representation



Application architecture



Application architecture



UI Design

Design goal: Create a simple and informative interface for users to learn about and interact with our model

- Technologies used: modified HTML/CSS template & turned into an angular application
- Design and front-/back-end development were worked on in parallel and later integrated together to become the final website
    - This method of collaboration allowed different team members to take ownership of different aspects of the final product and gain experience in merging code written by various people to create a product greater than the sum of its parts

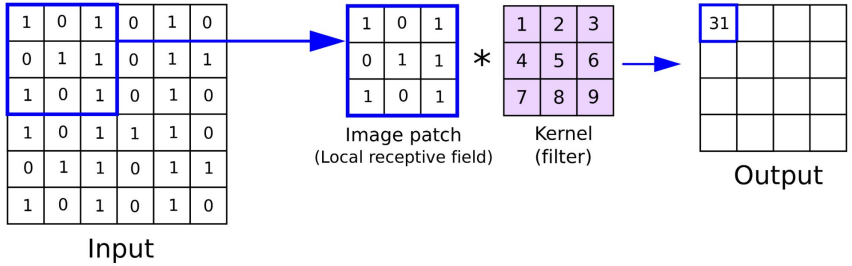
Technologies Used (Web-Development)

- Backend python server flask
  - Flask==1.1.2
  - Flask-Cors==3.0.10
  - Werkzeug==1.0.1
  - tensorflow==2.5.0rc1
  - opencv-python==4.5.1.48
  - numpy==1.19.4
- Frontend framework
  - Angular ^12.0.0-rc.0
  - Bootstrap ^4.6.0
  - Node 15.13.0
  - Express ^4.17.1

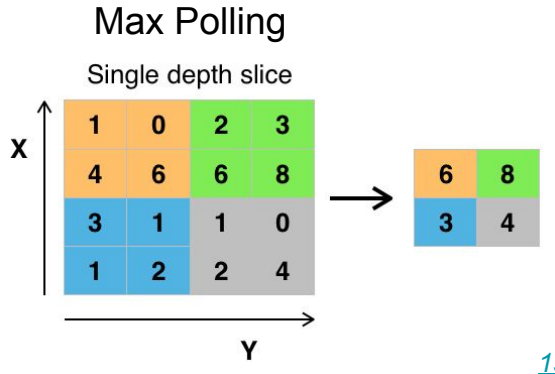
Technologies Used (Python)

Deep learning: convolutional neural network

- Decode images to numpy arrays
- Train the CNN model with numpy arrays and their labels
- Apply a filter over the image to extract the features



- Max Polling to scale down the image



[1st Image Source]
[2nd Image Source]

Development Methodology

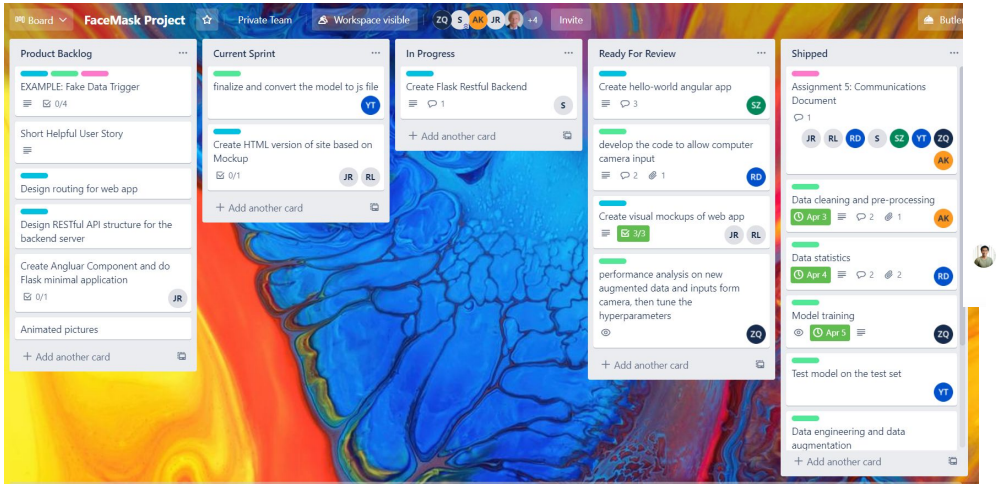
Date	Communication Type	Action Item			
23-Feb	Kick-off Meeting	Introduction between team members	Possible ideas discussion		
1-Mar	Project Team Meeting	Three new members joined since the last meeting	Introduction between team members	Possible ideas discussion	
3-Mar	Project Team Meeting	Finalize ideas brainstorming	Vote for project idea to pursue		
18-Mar	Project Team Meeting	Three new members joined since the last meeting	Explained the details of the project	Introduced similar project's reference	Sub-teams formed
25-Mar	Project Team Meeting	Setup repository	Scrum method adopted	Discussed how to code as a team	
31-Mar	Stakeholder Meeting	Meet the product owner			
1-Apr	Weekly Team Status Update	Setup a backlog with product owner	Sprint (start to code)		
8-Apr	Weekly Team Status Update	Weekly scrum meeting			
15-Apr	Weekly Team Status Update	First sprint ends	Deliver a demo to customers	Check backlog	Start second sprint
22-Apr	Weekly Team Status Update	Weekly scrum meeting	Final discussion	Presentation preparation	
29-Apr	Finalize Project	Final product delivery			

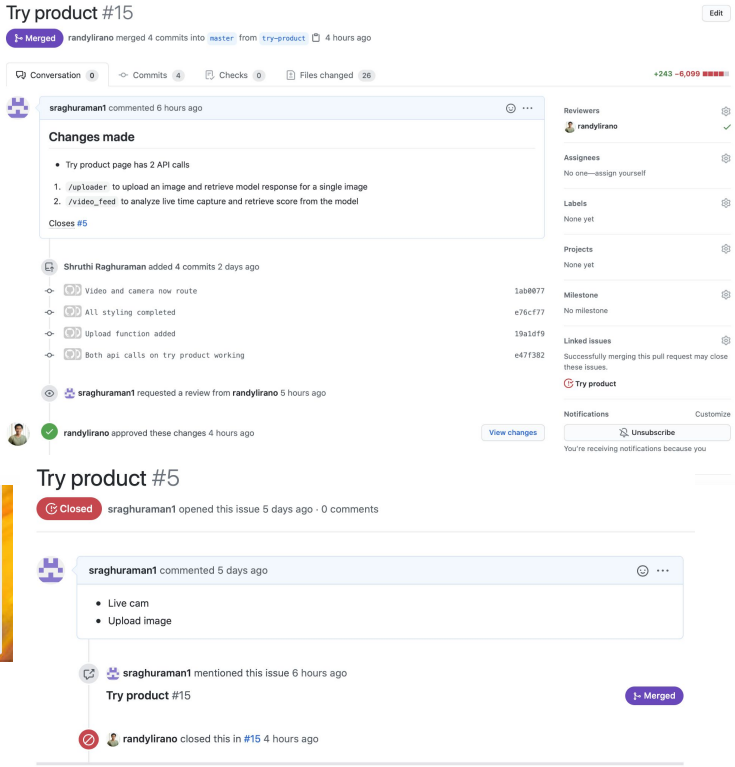
Scrum

- Web development and Python development team split
- Weekly scrum meeting and daily chatting
- Solid and continuous communication on Teams
- Requirements updated over time
- Motivated, self-organized individuals

Development Methodology Continued

- GitHub Code Sharing
  - Push, Pull, Merge, Issues
- Trello Task Management Board





Development Process

Python Team Early Process:

Data preprocessing

- Label preprocess

Data Augmentation

- Rotate the original data
- However, the augmented data is not used in this project
- In future, this data can be used for training and testing to reduce overfitting

Data statistic

Development Process

Python Team Later Process:

Model training

- CNN

Model testing

- Test on new dataset

Camera input

- Cv2 package

Demo delivered

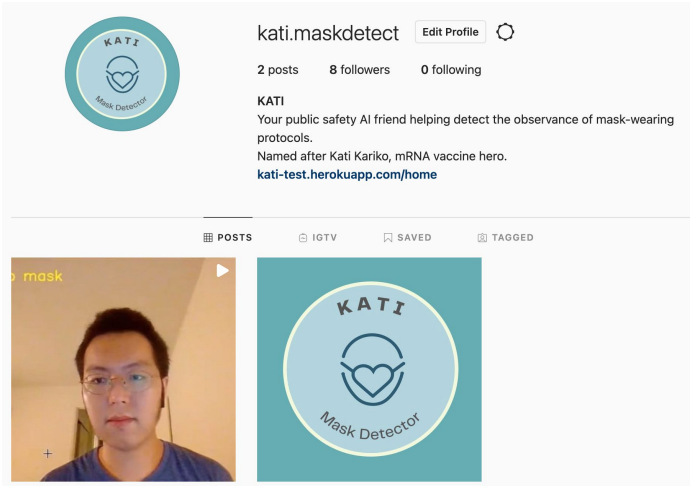
- Working Python standalone demo

Integrate with the Web team

Development Process

Web Team Development:

- Knowledge transfer
  - Tutorial
- Concept and web skeleton
  - Adobe XD
  - HTML
- Connect to Python
  - Backend Flask
- Implement frontend template & backend connection



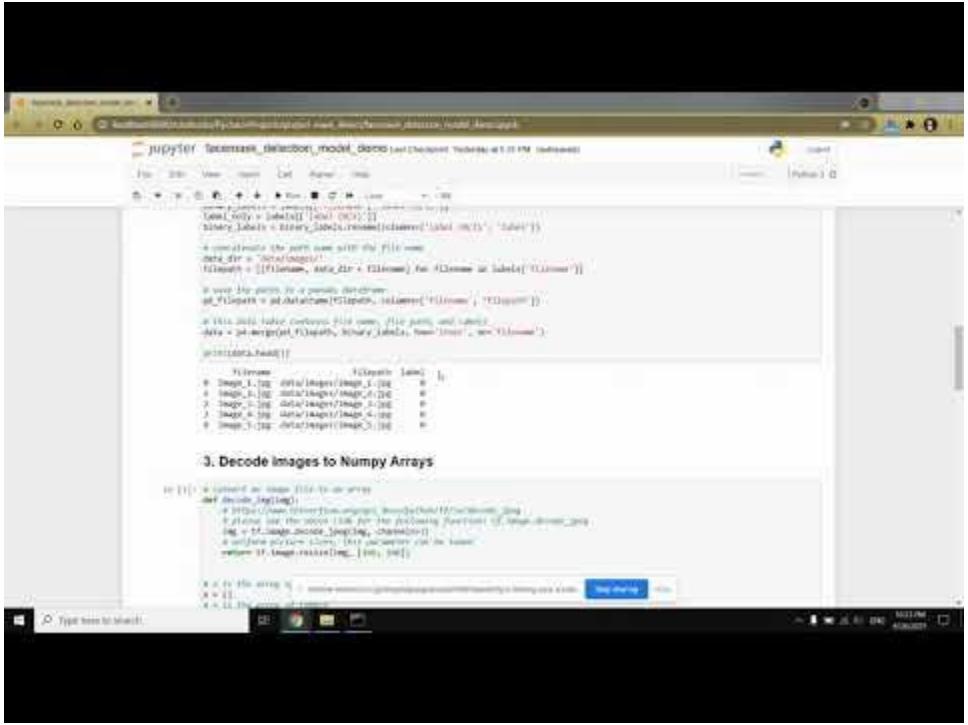
Social Media Development (@kati.maskfinder):

- To provide customers with updates on the tool
- KATI inspired by Dr. Kati Kariko, a scientist who laid the foundations for the vaccine

Organization of the Repository

- Our team uses GitHub for several reasons:
  - It allows different parts of the project being done in parallel through branching
  - It is easy to do version control
  - Github repository also enables Heroku deployment
  - README with team information and information on how to run/access project
- There are two main parts of our repository:
  - Main directory
    - Documents, Data set, Saved model, Data analysis notebook
    - Flask backend in api.py
  - Face-mask-ui folder
    - Angular front end
  - Kati-maskdetect-ui folder
    - Angular mock design

Project Demo - Python



Project Demo- Web Application

- How to run
  - Run requirements.txt
  - Start the Flask backend server in main repository
    - env FLASK_APP=api.py FLASK_ENV=development flask run
  - Cd into face-mask-ui
  - Start Angular front-end
    - Npm install
    - Ng serve --open

Thank you!

Acknowledgements:
Stakeholders Professor Gust and Luna Szymanski for their feedback and guidance