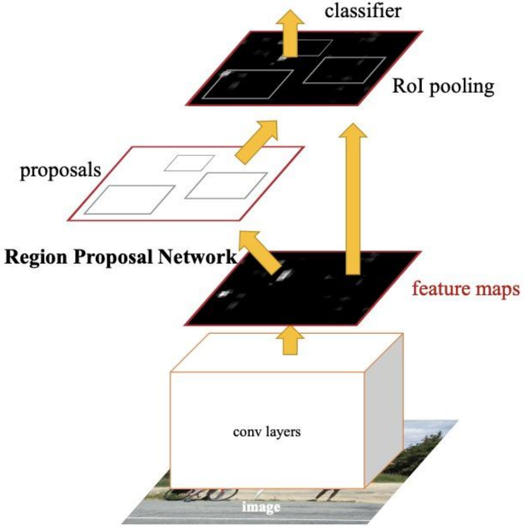
ELEN-6893
Big Data Analysis

Final Project:
Automatic Image Labelling System

Group #3: Jing Peng, Jiashu Chen, Yi Yang

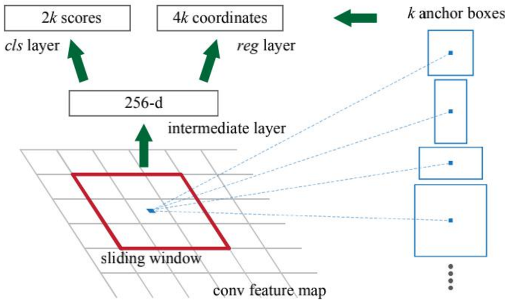
Date Performed: December 19rd, 2021

Methods - Algorithm

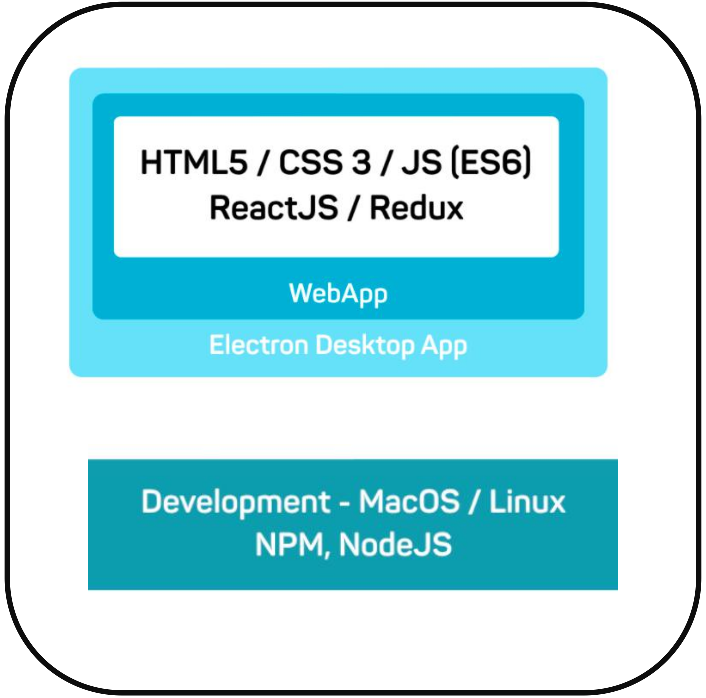


Faster-RCNN:

- Region-based classification and detection
- ResNet: Feature map extractor
- Region Proposal Network: Two-stage detector to generate region proposals
- Non-Maximum Suppression: Merge candidate region boxes
- RoI Head branches: Classification and Bounding-box regression
- Implement our experiments in PyTorch and visualize via OpenCV



Methods - Software



HTML5 / CSS 3 / JS (ES6)
ReactJS / Redux

WebApp

Electron Desktop App

Development - MacOS / Linux
NPM, NodeJS

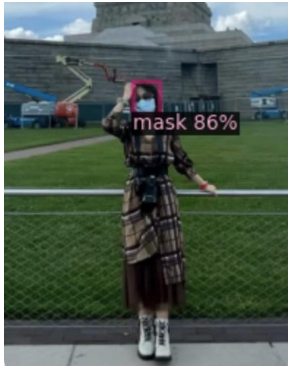
Across platform application:

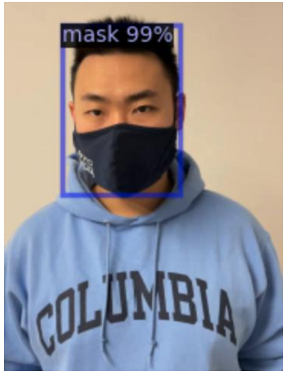
- Front-end:
  - Framework: Electron + React + Semantic UI
  - Data interaction: Axios + React Hooks
  - Package: NPM
- Back-end:
  - API: Flask
  - Deep-Learning Model
  - Database: MySQL8.0

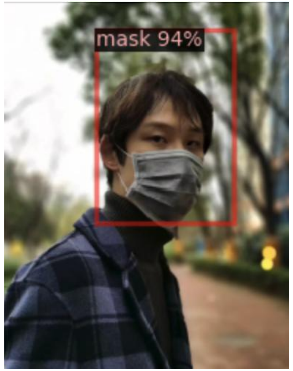
Current Results

ResNet34 + RPN:

Our model can detect targets in different colors, shapes and sizes



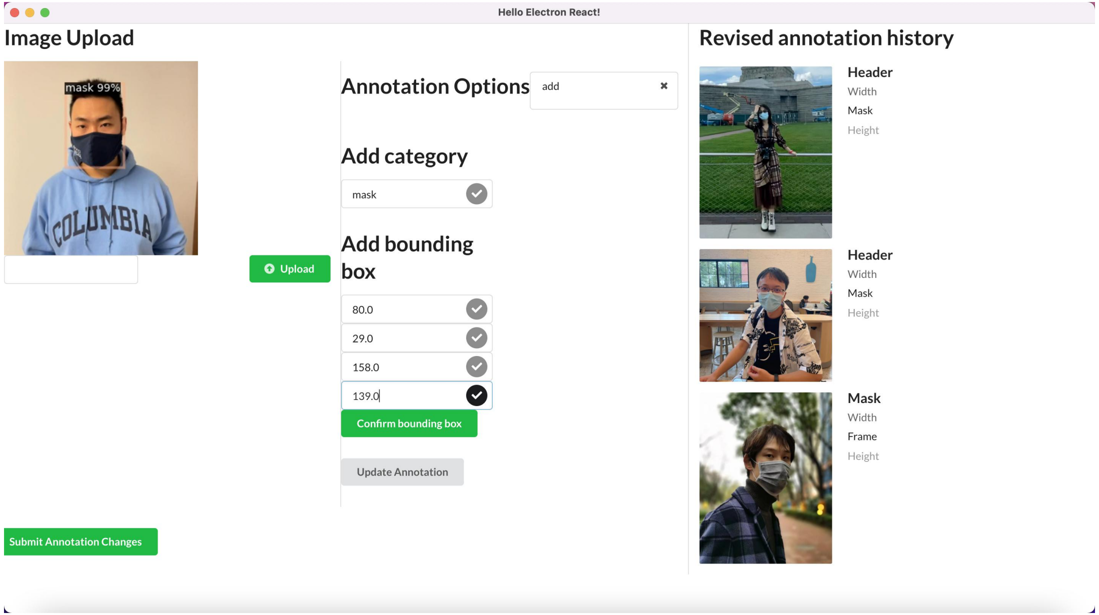




All training is done on GCP with one Tesla T4 GPU. Finished 20000 iterations in approximately 2 hours.

AP	AP50	AP75	APs	APm	APl
32.494	64.792	26.677	20.755	36.286	42.568

Current Results



Hello Electron React!

Image Upload

Annotation Options: add

Add category: mask

Add bounding box: 80.0, 29.0, 158.0, 139.0

Revised annotation history

Header
Width
Mask
Height

Header
Width
Mask
Height

Mask
Width
Frame
Height

- □ Upload your image to be labelled through our software user interface.
- □ Wait for annotations generated by our deep-learning model.
- □ Make modifications to achieve a better result if necessary.
- □ Confirm your labels and put it into training dataset.

Problems

- Model divergence
  - Non-ideal initiation
  - Adjustment on learning rate
  - Warm-up training
- Model Under-fitting
  - ResNet-18 seems to under-fit
  - Use a deeper net: ResNet-34/50
  - More epochs in the training process
- Lack of variety in training dataset
  - Detect some specific types of targets
  - Enlarge the training dataset
  - Ensure that training data contains targets of different shapes and color, etc.

Evaluation

- Deep-learning Model
  - Accuracy: AP as the most important criteria
  - Speed: How long it takes to process each image
  - Availability: Applicable to different targets and tasks

- Software
  - Performance: Using Lazyload for images and components in the page.
  - Functions:
    - Annotations of uploaded images
    - Change of the annotations
      - add
      - delete
      - update

Future Work

- Deep-learning Model
  - Model training function interface
  - Parameters tuning for better performance
  - Training dataset variety

- Software
  - Display real-time change of bounding box
  - Periodically training of the deep-learning model
  - Optimization of the front-end UI
  - Log system

Reference

- Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28: 91-99.
- He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.