AI Project Proposal

Mask Detection (Supervised Machine Learning)

Team Members

- JD Medlin (medlin.j@northeastern.edu)
- Xiangxin(Samuel) Ji (ji.xian@northeastern.edu)
- Gabriel Mendes (mendes.g@northeastern.edu)

Context & Problem Statement

For the sake of public health, we are looking to design a program that can, at a minimum, detect whether an individual is wearing a mask over their face. This undertaking would involve computer vision and machine learning. We may use the existing project for performance/accuracy comparison purposes in the finishing-up stage of development, but we are going to design the program from scratch for a deep understanding, as we think both of the fields involved are fairly applicable in our future practice. We may also add some utility features such as keeping track of people passing by, counting masked versus unmasked, for potential commercial use in buildings with people entering and exiting. The problem regarding the use of human labor or lack of occupancy regulation of masks would disappear. If time permits, we could then attempt to train our program to detect different types of masks (disposable, KN-95, etc.).

Things to Learn

All team members are relatively new to the topic of computer vision and machine/deep learning. We are going to review the official tutorial for OpenCV and research applicable machine learning algorithms. We may apply different mainstream machine learning algorithms on different datasets that we would like to differentiate between.

Dataset Sourcing

Kaggle will be the main source of the dataset for our project, particularly the Face Mask Detect dataset is promising for training faces with masks and there are multiple datasets of human faces that can be used for training faces without masks.

Libraries

We mainly need *OpenCV* as a support library to conduct face recognition, and *NumPy* will be used to build vectors for data classification. *PyTorch* / *TensorFlow* may be used if we end up with building a model that involves deep learning. Additional libraries will be referenced in the final report.

Relevant Links

- OpenCV Official Tutorial: https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html
- Project Blog: https://data-flair.training/blogs/face-mask-detection-with-python

Project Outcomes

Baselines

The program is able to label human faces and determine if they are wearing masks (after learning), with prediction confidence shown beside. Specifically, we will have a model-building program (implemented using a mainstream classification/supervised learning algorithm), and we'll be using the model to make predictions. The Model will be binary (un/masked) for the baseline and will be revised for stretch goals to be more specific in terms of masks types.

Stretch Goals (Reachable)

Try to compare with the existing project, and try to give fewer false negatives (where the program predicts individuals are wearing masks while they are not, the existing projects encounter this situation when the lower portion of the face is covered by ANYTHING, instead of only masks).

Stretch Goals (Challenging)

Build a more robust model that can extract the (intermediate) learning results from the binary labeling, and apply them to train a classifier that can be used to recognize different types of masks.

Very optional : Visualize the deep learning process.

Project Milestones & Important Dates

Milestone #1 (Mar. 25)

- Have a precise problem formulation with clear inputs and outputs.
- Finish learning and be comfortable using OpenCV and PyTorch functions.
- Gained the basic knowledge for AutoEncoder.
- Collect dataset for machine learning.
- The model can be built given a dataset, and the build is functioning.
    - The prediction can be defective at this point, the build should at least label human faces both with and without masks.

Milestone #2 (Apr. 12)

- Build a model that can precisely predict the binary (un/masked) input.
- Improve the model to conduct neural network learning (may be featuring AutoEncoder).
- Polish the UI and add features for practical application.

Presentation (Apr. 22 / Apr. 26)

- Presentation Demo Prep
- Presentation Scripts
    - High-level problem statement;
    - Problem formulation;
    - Methods/Algorithms applied;
    - Empirical results;
    - Analysis;
    - Application/Discussion/Conclusion

Reminder: Project Report (Apr. 29)

Week by Week Plans (8 Weeks Ahead)

Week 1 (Mar. 06 - Mar. 12)

- Run through the OpenCV tutorial.
- Slightly touch PyTorch basics
- Begin collecting a dataset to use for the project.

Week 2 (Mar. 13 - Mar. 19 [Spring Break])

- Decide the learning algorithm to use.
- By the end of the week, group members are comfortable using OpenCV and PyTorch functions.
- Start drafting/building the model.

Week 3 (Mar. 20 - Mar. 26)

- Continue building the model.
- Build a functioning trained model with the dataset
- Research neural networks concepts and fundamental implementations (PyTorch, Github, etc.).
- Conduct testings.

Week 4 (Mar. 27 - Apr. 02)

- Solve the final tasks from milestone #1, if there's any.
- Start working towards milestone #2.
  - Improve/change the learning algorithm so the model can provide more precise predictions.
  - If the minimum baseline is reached, enhance the model to do the deep learning.

Week 5 (Apr. 03 - Apr. 09)

- Continue improve the learning algorithm / enhance the learning model.
- Test rigorously and fix bugs.

Week 6 (Apr. 10 - Apr. 16)

- Begin creating Presentation, plan to finalize program.
- Test rigorously and fix bugs.
- Compare the finalized program with the existing project and analyze the performance/accuracy.

Week 7 (Apr. 17 - Apr. 23)

- Begin Drafting the final report.
- Finalize presentation and present.

Week 8 (Apr. 24 - Apr. 29)

- Finalize final report.