Proctor exams using AI

Group Members:
Ruohua Li(lir11@rpi.edu)
Fengwei Liu(liuf7@rpi.edu)

Motivation and Research:

The primary motivation of this project is to utilize machine learning techniques to solve a real-world problem: proctoring exams. Prior to actual implementation, we did extensive research on this area, and there are many researches existing on different aspects. Extracted head positions from the detected head in the video could potentially find abnormal head movement[1]. Posture analysis could also be a solution where the algorithm could capture potential cheating behavior from different postures of students[2]. It is also a challenge to integrate different models together. There is also research where multiple webcam plus microphones are being used to find different features such as sound, gaze movement, active window detection, and more[3]. To simplify our solution but not compromise its effectiveness, we decided to adopt our understanding of computer vision and create a pipeline to realize our goal. The pipeline essentially contains two parts: object detection and facial emotion classification. And we will address more details below.

Pipeline Structure:

Object Detection

The goal for this task is to accurately locate the human face along with other abnormal objects (such as smartphones) that shouldn't exist during an exam given an image or a video footage.

Dataset:

1. Wider Face Dataset[4]
   This dataset was selected based on the face detection benchmarks[5].
   a. This dataset has 32,203 images with 393,703 faces. It has been divided to 40% train, 10% validation, and 50% test (80% train and 20% validation if we consider the test set is infinite).
   b. Since this is a benchmark dataset, minimum preprocessing is performed. After preprocessing, there are 7366 training images (79.87%) and 1856 validation images (20.13%). Some images have been removed since the bounding boxes of faces are too small which only have a few pixels of area.
2. Dataset from open source project on roboflow[6]
   These datasets were chosen as we also want to detect abnormal objects.
   a. This dataset has 1018 training images (88.44%) and 113 validation images (11.56%).
   b. The dataset does not have missing labels and is already in YOLOv5 label format. No preprocess was done on this dataset.

The YOLOv5 automatically calculated and applied class weights for this imbalanced dataset.

Model:
After consideration, we chose pre-trained YOLOv5[7] as our model based on the following requirements:

1. Be able to perform real-time object detection tasks (and be able to take videos as input).
2. Have a high score on the object detection benchmarks[8].
3. Easy to implement.
4. Fast training speed.
5. Be able to handle imbalanced dataset.

Result:
The mAP@50 after 20 epochs is 93.9% for Face and is 98.3% for Mobile.

Facial Emotion Classification

The primary purpose of this section is based on the paper “Automated Cheating Detection in Exams using Posture and Emotion Analysis”[2]. In this paper, they propose a solution that uses a combination of emotion analysis and posture examination. Therefore, we decided to adopt the idea of emotion analysis and implement it into our pipeline.

Dataset:
We initially decided to use fer2013[9] as our dataset solely. We searched for other datasets, such as AffectNet[10] and Dreamer[11]. However, those datasets are either too large for us to process (120+GB) or not publicly available for us to access. Therefore, we settled with fer2013 since this dataset is well-studied and tested.
In order to achieve a better result (around 70% testing), we did some research about how others could achieve a high testing result on the same dataset[12]. The most common technique is using a pre-trained model adopted from other datasets. Therefore, we decided to add another dataset, VGGFace2[13], to pre-train our model.

Model Selection:
We first tested resnet18, resnet32, resnet50, and VGG networks. However, the results are not ideal as most of them cannot converge until 50 episodes. The validation accuracy cannot be improved when reaching around 60%. The major limitation, we believe, is that our model can hardly capture the subtle differences in each emotion. Therefore we adopted the transfer learning technique where we first load a pre-trained resnet50 model previously trained on VGGFace2.[14] Then, we changed the conv1 layer and eliminated the max pooling layer. Since the original resnet50 will downsample an image by half, we don’t want to lose too much information during downsampling,

considering the size of fer2013 images. We also changed the last fc layer to perform the classification task as we needed.

Result:
Overall, our facial emotion classification model could achieve around 71% accuracy, which is good enough for our task.

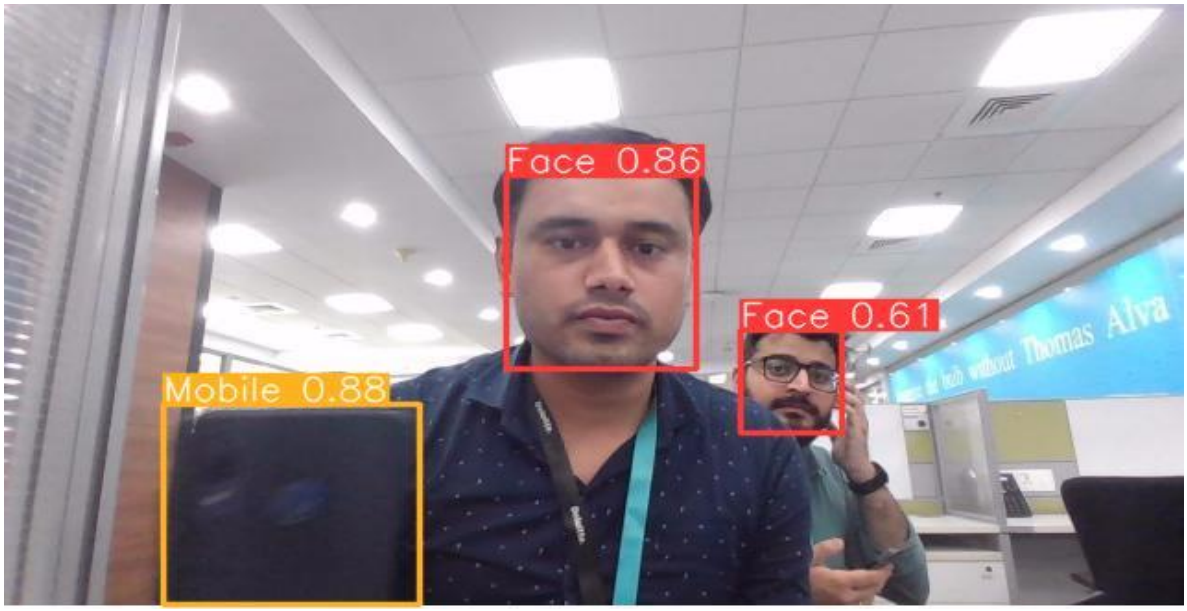
Pipeline Evaluation:

The lack of a public exam-taking and cheating-attempting video dataset was one of the challenges of this project. Therefore, we can only evaluate our pipeline separately. However, we did achieve a decent result on each individual part, as discussed above. On the other hand, our model is highly extensible. This means that we could add more modules, which contain a different analysis of other features, in the future.

Pipeline Inference:

The image is first passed to YOLOv5 to detect and crop faces and abnormal objects (a confidence threshold of 0.6 is set based on the F1-Curve from the training result). Then faces will be gray-scaled and resized to 48x48 for facial emotion classification. A report regarding the result will be made and images of cropped objects will be saved to the same directory.

Sample



Face 1: Fear

Face 2: Neutral

Abnormal object

Reference:

[1] https://arxiv.org/pdf/2101.07990.pdf
[2] https://ieeexplore.ieee.org/document/9198691
[3] http://cvlab.cse.msu.edu/pdfs/Atoum_Chen_Liu_Hsu_Liu_OEP.pdf
[4] https://doi.org/10.48550/arXiv.1511.06523
[5] https://paperswithcode.com/task/face-detection
[6] https://universe.roboflow.com/yolo5-kjv0a/tustrain
[7] https://doi.org/10.48550/arXiv.2107.08430
[8] https://paperswithcode.com/task/object-detection
[9] https://www.kaggle.com/datasets/msambare/fer2013
[10] http://mohammadmahoor.com/affectnet/
[11] https://zenodo.org/record/546113#.Y5Knv3bMJGo
[12] http://cs230.stanford.edu/projects_winter_2020/reports/32610274.pdf
[13] https://arxiv.org/pdf/1710.08092.pdf
[14] https://github.com/cydonia999/VGGFace2-pytorch