AI Invigilation System Documentation

COMPSCI 4ZP6 group#3: Zicheng Guo, Xing Li, Yongxin Zhao

Department of Computing and Software, McMaster University

Supervisor: Dr. Rong Zheng

1. Project Description

1.1 Project Perspective

Currently, most exams are supervised by real human-invigilators. Usually, for an exam with around 300 examinees, only 2-3 invigilators are involved. As a result, many students tend to cheat under such conditions. There are a few similar systems on the market, but they all target online exams or a computerized exam center. Our product will target in-person exams either at a high-level educational institute or professional exam center.

1.2 Project Scope

The AI Invigilation system helps the supervisors of in-person exams by setting up exam timers, detecting and recording suspicious behaviors of examinees during the exams. For some large-scale examinations, it's hard for the supervisors to pay attention to each and every examinee. At the same time, detecting cheating behaviors usually requires a period of time to observe that specific suspicious examinee, which makes it harder to fairly supervise all examinees. The AI Invigilation system can monitor several examinees at the same time and once their behaviors meet some pre-set criterion, the corresponding examinees will be highlighted with explanations for the supervisors to make further decisions. Besides, the criterions and factors are customizable for different examinations by the supervisors.

The system will store the videos on local machines to reduce disputes in the future.

1.3 System Functions

Before the exam:

- The system shall allow supervisors to log in and authenticate their identities.
- The system shall allow supervisors to input exam information and set triggers.
- The system shall display the exam information based on the previous user input.
- The system shall display a GUI that allows users to start and end sending camera feeds to the BE.

During the exam:

- The system shall display camera feeds on the screen which is the one that only the invigilators can watch.
- The system shall detect suspicious behaviors and highlight suspicious examinees based on the pre-set criteria.
- The system shall send notifications with reasons and show them on the interface once suspicious behavior is detected.
- The system shall record the camera video once the exam is started.

After the exam:

- The system shall end recording automatically at the end of the exam.
- The system shall generate a report including exam information and timestamps of detected suspicious behaviors when the exam is ended.
- of Suspicious Behaviour in the Examination Hall.

2. User Instructions

2.1 WI Instructions

Setup Environment

- Start react app: npm start
- Start server: run AI-invigilation-system\AI_Invigilation\Auth.py

User Authentication

- Select user type by clicking the corresponding button

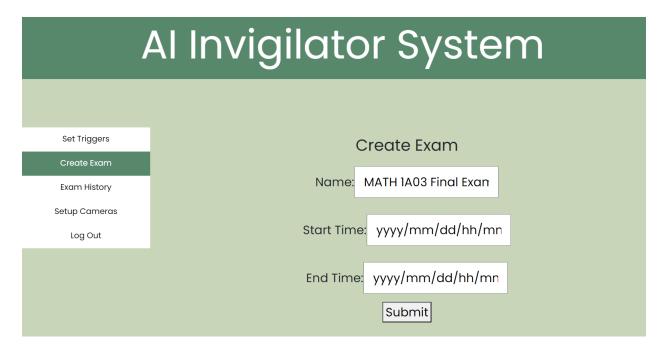# AI Invigilator System

User Authentication

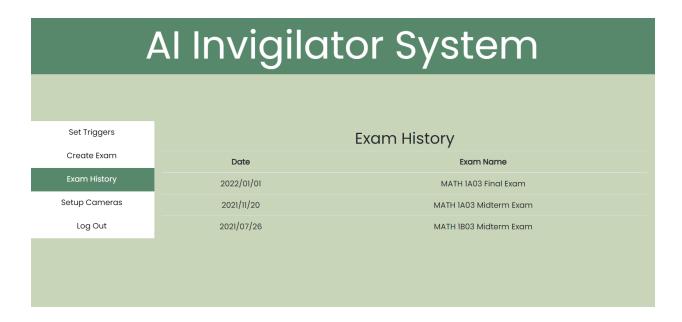Supervisor

Staff

- Enter user confidential

AI Invigilator System

Supervisor Authentication

Institution ID : [_____]   Send Code

Code : [_____]   Log In

- Active / deactivate triggers by selecting/unselecting the checkbox, click submit to apply the change

AI Invigilator System

Set Triggers

Trigger Type	Parameters	Status
Head Position Trigger	is mouth open	☑
Head Position Trigger	head yaw	☐
Head Position Trigger	head pitch	☑

Submit

● Entering exam information to show a info page

AI Invigilator System

Set Triggers

Create Exam

Exam History

Setup Cameras

Log Out

Create Exam

Name: MATH 1A03 Final Exan

Start Time: yyyy/mm/dd/hh/mn

End Time: yyyy/mm/dd/hh/mn

Submit

● Display past exams history

- Active / deactivate cameras by selecting/unselecting the checkbox, click submit to apply the change

- Logout the current account by clicking the button

2.2 BE Instructions

Setup Environment

- Install Environment using conda: conda create --name <environment_name> --file back_end\project_conda_env.yml
- Verfiy CUDA 11.6 and cudnn for 11.6 is installed properly
- Activate the conda env just created
- Switch to directory back_end and run python main.py using the above environment

Arguments

- –local [bool] a boolean value wether the backend run as online mode(using MM as source of camera) or local mode(using the camera from the machine running BE for debug purpose)
- –backup_dir [path] a path to where the frame been backup
- –port [int] port for listening
- –ip [str] ip for listening, usually set to 0.0.0.0

If all above setting is done, once the backend is running, it will immediately start listening income traffic:


```
Hash saved to ./hash.txt
Online mode, start listining on port 8765 ...
No connection attemp 1, retrying after 10s ...
No connection attemp 2, retrying after 10s ...
No connection attemp 3, retrying after 10s ...
No connection attemp 4, retrying after 10s ...
```

2.3   MM Instructions

Setup Environment
- Installed required package for MM, for simplicity, you can use the same environment for BE but without CUDA/torch and related library.
- Run python main.py to startup GUI

Usage
- Run the software once and you can see the config file generated under the same directory.
  - BE_Network:host: the host IP address of BE machine
  - BE_Network:port: the port of listening of BE machine
  - File_path:recording_path: the directory where recording will be saved
  - File_path:report_path: the directory where the report will be saved
  - Camera:black_list: the id(s) of camera that will not in use.
- Once configuration is saved, you can use the GUI:
  - Start BE: make connection to BE and start streaming frames to backend for detection
  - Stop BE: release all camera and kill the connection
  - Start WI: same as `run AI-invigilation-system\AI_Invigilation\Auth.py` in WI instruction
  - Cam: Open another window for viewing camera feeds

3.   Project Architecture

3.1   WI System

System main page

Path: /

File:

- ai-invigilation-system/src/components/Auth.js
- ai-invigilation-system/src/components/Auth.css

Component: MainPage

Description: This page is the main page of the system, it provides two links to the separate authentication page for supervisors and staff members.

Supervisor authentication page

Path: /auth-supervisor

File:

- ai-invigilation-system/src/components/AuthUser.js

Component: AuthSupervisor

Description: This page is the authentication page for supervisors, it handles user input (institution id and access code) and authenticates users. Once the supervisor is authenticated, it redirects to the supervisor's home page.

Staff authentication page

Path: /auth-supervisor

File:

- ai-invigilation-system/src/components/AuthUser.js

Component: AuthStaff

Description: This page is the authentication page for staff members, it handles user input (institution id and access code) and authenticates users. Once the staff member is authenticated, it redirects to the staff home page.

Supervisor home page

Path: /supervisor-home

File:

- ai-invigilation-system/src/components/UserHome.js
- ai-invigilation-system/src/components/UserHome.css

Component: SupervisorHome

Description: This page is the main page for supervisors, it contains a sidebar on the left of the page and a content section. The sidebar can navigate users among available features, and the default page is set triggers.

Staff home page

Path: /staff-home

File:

- ai-invigilation-system/src/components/UserHome.js
- ai-invigilation-system/src/components/UserHome.css

Component: StaffHome

Description: This page is the main page for staff members, it contains a sidebar on the left of the page and a content section. The sidebar can navigate users among available features, and the default page is set up cameras.

System features (inside user home page)

1. Set triggers

Path: /supervisor-home/

File: ai-invigilation-system/src/components/user/SetTriggers.js

Component: SetTriggers

Description: This page is only available for supervisors and is the default page in supervisor home page. It contains a table displaying each trigger, users can enable/disable any triggers by checking/unchecking the checkbox in the status column.

2. Create exam

Path: /supervisor-home/create-exam

File: ai-invigilation-system/src/components/user/CreateExam.js

Component: CreateExam

Description: This page is only available for supervisors. It contains a form that receives user input and will pop up a new window displaying the exam info once the user saves the exam (submit the form).

3. Exam history

Path: /supervisor-home/exam-history

File: ai-invigilation-system/src/components/user/ExamHistory.js

Component: ExamHistory

Description: This page is only available for supervisors. It contains a table displaying each exam and links to the recording and reports of the exam stored in the local machine.

4. Setup cameras

Path:

- (for supervisors) /supervisor-home/setup-cams
- (for staffs) /staff-home/setup-cams

File: ai-invigilation-system/src/components/user/SetupCams.js

Component: SetupCams

Description: This page is available for both supervisors staffs while the two different paths redirect to the same page. It contains a table displaying each camera, users can enable/disable any cameras by checking/unchecking the checkbox in the status column.

5. Log out

Path:

- (for supervisors) /supervisor-home/log-out
- (for staffs) /staff-home/log-out

File: ai-invigilation-system/src/components/user/LogOut.js

Component: LogOut

Description: This page is available for both supervisors staffs while the two different paths redirect to the same page. It contains a link that logs out the user and redirects to the main page of the system.

Exam info page

Path: /exam-info

File: ai-invigilation-system/src/components/UserHome.js

Component: ExamInfo

Description: This page will be created once the user creates the exam (submit the create exam form), and it displays the information and a countdown of the exam remaining time.

3.2   BE System

AlgorithmControler

File: AI-invigilation-system\back_end\src\AlgorithmControler.py

Component: AlgorithmControler

Description: This module controls and manipulates all detection algorithms.

- **inference()** will take in a PIL image and run the detection algorithm and prase the raw result into our detection features.
    - Tricks: Original plan is to use multiple computer vision models to detect different features, but since we do not require very accurate results, some of them can be replaced by others. E.g. for head yaw degree estimation, instead of having a separate model to estimate it. We can use the result from landmark(i.e. The distance between left boundary and right boundary to the nose to estimate the head position). Similar to head pitch estimation

CommunicationManager

File: AI-invigilation-system\back_end\src\CommunicationManager.py

Component: CommunicationManager

Description:This module using socket to implement communication between BE and MM.

- accept_connection() will accept an incoming connection to the BE.
- send_message(message) will send message to the connection thats has been make through accpt_connection.
  - Message has been divide into two type: STR and IMG, which are string and image.
- receive_message() will receive message from the connection thats has been make through accpt_connection.
- close() will close the connection and the socket

StorageManager

File: AI-invigilation-system\back_end\src\StorageManager.py

Component: StorageManager

Description: This module controls and writes images to file.

- **saveFrame()** will take in a cv2 image and save it to present directory as a backup.

TriggerManager

File: AI-invigilation-system\back_end\src\TriggerManager.py

Component: TriggerManager

Description: This module is a singleton class that stores and manipulates triggers.

- **set_trigger()** will set the trigger to some threshold.
- **get_trigger()** will get the threshold for the trigger.

utli

File: AI-invigilation-system\back_end\src\utli.py

Component: cv2_2_pil pil_2_cv2 bb_2_rect dist_2d mid_point_2d sigmoid

Description: utility functions for convience.

- **cv2_2_pil()** will convert a cv2 image class to a PIL image class.
- **pil_2_cv2()** will convert a PIL image class to a cv2 image class.
- **bb_2_rect()** will convert a boundingbox (i.e. a numpy/python list of length 4) to a dlib rectangle class.
- **dist_2d()** will get the euclidean distance of 2 points.
- **mid_point_2d()** will get the middle point of 2 points.
- **sigmoid()** standard sigmoid function.

torch_mtcnn_custom

File: all file under torch_mtcnn_custom folder

Description: a custom version of https://github.com/TropComplique/mtcnn-pytorch

- This is a custom version of python package mtcnn-pytorch, adjustment has been make to suit our need.
- The interface is remain original
- Some change been made: changed some removed function for pytorch, changed CUDA setting. Some numpy torch tensor transform is added to suit newer version.

3.3   MM System

Exam_Manager

File: ai-invigilation-system/AI_Invigilation/Exam_Manager/exam_manager.py

Component: Exam Manager

Description: This module will be helper functions for all other modules.

- **generate_report()** will compose to **end_exam()** and **end_exam()** will be composed to another function in another module as one component. This function outputs the suspicious clips timeframe out as a txt file.
- **set_info ()** and **display_info()** will be components for **start_exam().** These two functions set up the basic information of the exam and display it.

- **start_exam()** and **end_exam()** will be helper functions for module video_controller.

Student

File: ai-invigilation-system/AI_Invigilation/Exam_Manager/student.py

Component: student

Description: This module defines the class student, with id(int) and suspicious(boolean) as features.

system

File: ai-invigilation-system/AI_Invigilation/Exam_Manager/system.py

Component: Exam Manager, system

Description: This module does generate hashcode for users and authentication.

- **generate_supervisor()** function does generate a hashcode for users that are supervisors of the exam.
- **generate_staff()** function does generate a hashcode for users that are normal staff of the exam.
- **authenticate_supervisor()** function authenticate the hashcode of supervisors.
- **Authenticate_staff()** function authenticate the hashcode for staff.

Message

File: ai-invigilation-system/AI_Invigilation/Exam_Manager/system.py

Component: student, message

Description: This module imports the student as an object.

- **show_message()** does interact with the WI system, when you click on the button of the flagged person, it will show the possible reasons that trigger the flag.

- **hide_message()** does interact with the WI system, clicking on the button with a hide label on it will conceal the message of the trigger.
- **dismiss_message()** does interact with the WI system, when a possible examinee keeps triggering the flag.

trigger

File: ai-invigilation-system/AI_Invigilation/Exam_Manager/trigger.py

Component: trigger

Description: This module creates class trigger

- **active_trigger()** does interact with the WI system and BE system, when you click on the box of the certain trigger, it will activate the trigger you selected and pass the information through the MM system back to the BE system.
- **cancel_trigger()** does the same thing but totally reverse than the active_trigger(), you can unbox the certain trigger, it will deactivate the trigger you selected and let the BE system know though the MM system.

video_controller

File: ai-invigilation-system/AI_Invigilation/Exam_Manager/video_controller.py

Component: exam_manager, video_controller

Description: This module controls the class video_controller.

- **start_recording()** will start recording the footage of the camera and start the exam. It composed the function from exam_manager, including start_exam(), set_info(). The function will ask several inputs from the WI interface.
- **end_recording()** will end recording and use the function from exam_manager end_exam() to end the exam, which will save the crucial time points into a txt file to the local machine.
- **send_Next_detection_frame()** will send the camera image back to the BE system from the camera, and send back the result obtained from the BE though MM system back to the WI system.

camera_controller

File: ai-invigilation-system/AI_Invigilation/src/camera_controller.py

Component: exam_manager, video_controller

Description: This module controls the class camera_controller.

- **start_recording**() will start recording the footage of the camera and start the exam
- **get_current_frame**() will return a copy of the current frame.
- **stop_recording**() will end recording and set self.recording status to be false.

Convert_to_txt

File: ai-invigilation-system/AI_Invigilation/src/convert_to_txt.py

Component:None

Description: This module controls the class convert_to_txt

- **convert_json**() will read the raw file from output of the module and convert the json file to a dataframe.
- **transform_data**() will using the dataframe and do a series of data processing, including read the datetime from the file name and put it in a column, other than that, it will read that column and calculate its average diff that triggers the system to put it in a now column called 'average trigger time'.

action_queue

File: ai-invigilation-system/AI_Invigilation/src/action_queue.py

Component:None

Description: This module controls the class action_queue

- **add_action**() will allow users to add action into a queue, append it to self.queue.
- **get_top_action**() will return the top action within the queue.

- **remove_action()** will allow users to remove top action within the queue.
- **is_empty()** will check if the queue is empty or not.

connection_manager

File: ai-invigilation-system/AI_Invigilation/src/connection_manager.py

Component:storage_manager

Description: This module controls the class connection_manager

- **connect()** will connect BE system.
- **start_sending()** will get the next detection frame and send it to BE, as well as save the information that triggers the system to storage_manager.
- **start_sending()** will end sending messages to BE.
- **receive_message()** will receive messages from WI and process it.