



Home Examiner

The Problem

For my challenge I choose to tackle the problem of online examination.

During the recent covid-19 outbreak online examination has become an important process for many universities, because physical educational hasn't been allowed in many countries. Also before the covid-19 outbreak, online examination was already being used in some companies and universities, because it allows people to take exams, regardless of where they are in the world.

But major adaption of online examination in educational instructions could also be used for more personal education to take place. For example, at the beginning of this year I spoke with the dean of my current BSc. Chemical Engineering in Delft, the Netherlands. We talked about how every student learns at a different speed and some could finish a certain course in a week, while others would need 6-8 weeks. She argued that scheduling an exam and necessary exam-room would be the hardest part of implementing this new way of personalized testing. Therefore adopting online examination as a standard would create more opportunities for personalized examination.

However, both students and universities experience some problems regarding online examinations, namely:

- Universities find it hard to make sure that students don't do any cheating during an online exam. Therefore some universities use online surveillants or collect webcam video footage of students making the exam and scan this through a cheat-detector. The first approach is quite problematic, because most universities don't have enough surveillants to monitor all their students. The second approach is also problematic because many students are having concerns about their privacy, because the universities are storing video footage of them working at home.
- Students do sometimes experience disadvantages from making online exams instead of physical ones. For example, some universities implement time limited question availability to avoid cheating, because they don't have an better alternative to monitor students cheating. This way students experience a lot of time pressure to finish each question in time, while on a regular exam they can plan their questions over the time however they would like.

Therefore I think there should be a better solution for online examination, which allows universities to monitor if students are cheating and provides no disadvantages for the students taking the online exams. This must all be done, while taking the students privacy into account and therefore no video footage of students should be collected or stored.

Motivation

I chose this problem first of all because some fellow student friends of mine encountered disadvantages because of online examination. Also a lot of students in the Netherlands actually have their exams rescheduled because their university doesn't have an online examination strategy, which is of course not very fair to them. But besides that, I am also very excited to see my old bachelor's program adopt this new way of personalized testing and maybe even more universities will then implement this strategy, because online examination can be adopted widespread.

When identifying this problem I mostly spoke to fellow students that I know in the Netherlands, but I also checked some articles on the internet about students and government's concerns about privacy, safety and effectiveness regarding online examination

So in order to make an online examination strategy, that detects cheating of students and is easy to use for students, I decided to write some demo software in order to demonstrate how I envision something like this.

The solution

First of all, one major requirement of this program is that no video footage is stored by institutions, so that students won't have to worry about their privacy. Therefore I decided that all video footage used to detect potential cheating should be processed locally on the students computer and only when a 'cheating signal' is detected, this signal should be send to the institution. Also I assumed that all exams will be made with pen and paper.

In order to detect these cheating signals, I figured that there are 2 major ways of cheating during an online exam: 1. Non-allowed use of information/tools/communication on your computer, such as google, WhatsApp web, wolfram-alpha etc. 2. Non-allowed use of tools/communication in your examination environment, which will be your desk.

1. Non-allowed use of information/tools/communication on your computer

To tackle this first problem, I figured that when this examination software launches, the students should only be able to use this software and non-other tools on their computer. Therefore I implemented the following features in the software:

- When the software starts, it launches a kiosk mode screen of chromium, this way the student will only be able to see this webpage and have no buttons to click the page away.
- Of course somebody could still use their windows key + tab to leave the chromium webpage and open another program. Therefore I decided that when the software starts, first of all a bash script will kill some processes running, like chrome or internet explorer. But most importantly, when the student starts his exam in the software, input from their mouse and keyboard will be blocked temporarily, until the exam is over.

2. Non-allowed use of tools/communication in your examination environment

To tackle the second problem I figured that students could basically only cheat using communication devices like smartphones and tablets or do identity fraud. I assumed that for higher-education exams, study books and notes are allowed, because the exam questions are generally more problem solving questions(at least, that's what they should be).

My first thought was that the students environment could be analyzed by using the student's webcam and some AI/Computer Vision. But while trying to make this work, I realized that I should also add another camera angle pointed at the students desk, because you should actually be able to make sure there is no identity fraud and no use of non-allowed tools simultaneously. So I came up with the idea to use your phone's camera for that, because every student definitely has a smartphone available!

So we have the video stream from the pc webcam faced at the student's face and the video stream from the smartphone cam faced at the students desk. For aiming the phone's camera at your desk I suggest students will purchase a flexible phone holder for about 10 euros, like this one:



To accomplish the goal mentioned above, I implemented the following features in the software:

Verification of student/environment

- First check if the phone and pc camera are actually on
- Secondly check if the person behind the laptop is actually the student, using face recognition
- Thirdly check if the person's hands are on their desk(which means they are sitting behind their desk)
- Finally check if an audio recording of both the phone and pc are approximately equal in loudness, so that we are sure the phone camera and pc camera are in the same room, filming the same desk

Cheat detection during exam

- Use the PC camera to check for a different face than the student(identity fraud) and check for presence of smartphone

- Use the phone camera to check if the person's hands stay on the desk(If they are not the student is not there or cheating) and check for presence of smartphone
- The above mentioned events are considered forms of cheating and will be stored in the exam logfile

The Home Examiner is now just a demo version, which means that the students list/ pictures, exams and logfiles are now stored locally. In an actual working version, the students information, the exam and the specific time for the exam will be retrieved from the school software system. Also the event logfiles with the cheating signals will be send to the schools and they decide what would be classified as cheating.

I do think this approach of online examination has the potential to be the best solution regarding privacy and academic integrity. However, I do think that other computer vision models than the ones used in this project could greatly improve the concept. I do also think that the back-end of the program should perhaps be deployed on a virtual machine/cloud platform, so that there is more CPU and GPU power for the models to run, while these remote machines would still not store the video footage.

Personal Progress

This was my first project working with computer vision, which I am now much more familiar with in terms of which libraries to use, how to make a good dataset and how to train and test models.

In this process I actually found some very useful pre-trained computer vision models, which are mentioned on the GitHub page, but for the smartphone detection I couldn't really find any. So I web scraped around 1700 pictures of smartphone laying on desks and paid someone on Fiverr to annotate the images for around 9 dollars. Then I tried training a model on this dataset with 5 different libraries/frameworks, but this turned out to be quite hard(mostly because the dependencies and libraries only worked for certain combinations of versions). But in the end I did manage to actually train a model on a friends Nvidia GPU so that was a pretty cool achievement.

In my past coding projects I have never really focused on building a good looking interface, because I do value functionality a bit over looks. But for this project I really wanted to present something great looking and therefore I also really progressed in my web design skills.

But I think the biggest progress during this exercise was more about my coding workflow. I have actually never had to finish a project like this in just 3 weeks. So were in the past I usually just dive into the first solution for a certain problem I have, I now found out that it can be way more efficient to first consider a range of different possible solutions for the problem. After that you can choose the approach that you think is the best and most likely to succeed. For example, I started making an android app to capture the phone's camera and apply the computer vision models on it, but after 2 days of struggling I realized that I could rather just capture the video stream on the phone and then process it on the PC app. Than I found that such an app already exists, namely IP Webcam. So if I had first considered all different options I might have come to this same solution way quicker.

The future

For this project I choose a more general approach of the problem, but let's imagine for fun that this project would be continued by a group of Code students. These would be the challenges that different people could work on:

- Software Engineering: Improve the Computer Vision models to detect objects more accurate and make them more lightweight(so that no laptop will run out of memory)
- Interaction Design: Improve the interface, so that students are even less likely to cheat
- Product Management: Identify requirements from educational institutions to implement this software and implement those in the software