CSE 40537/60537 Biometrics - Spring 2022
Instructor: Daniel Moreira (dhenriq1@nd.edu)

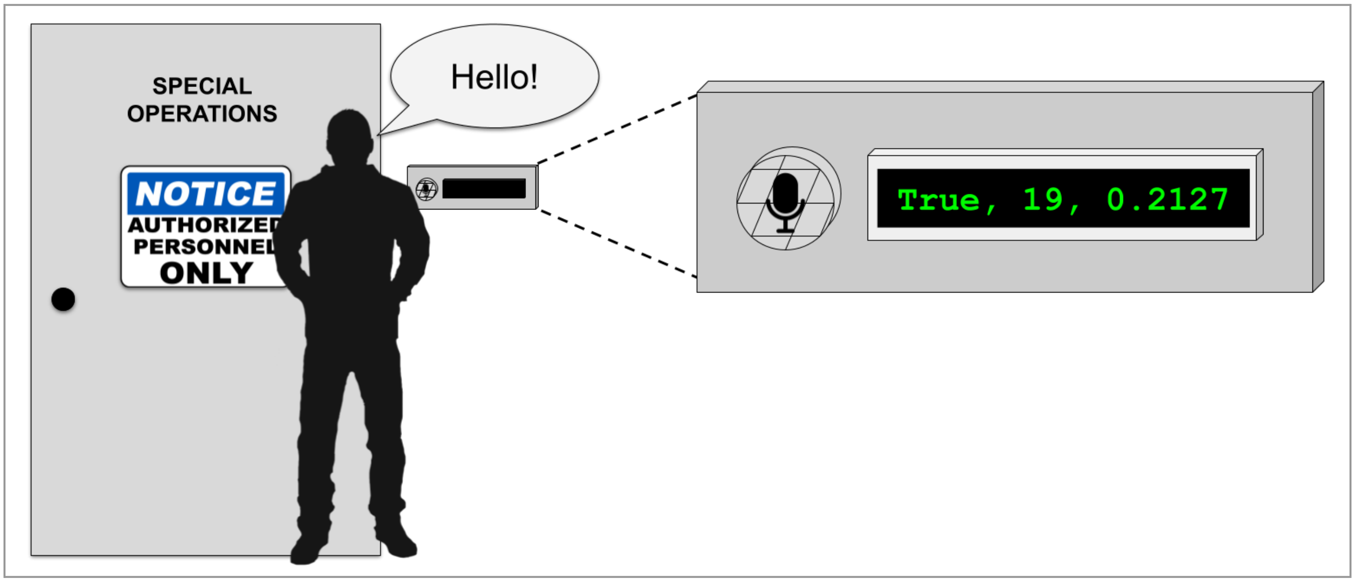
Student (Printed): ______________________________ **(Signature):** ____________________

Final Exam - 05/05/2022

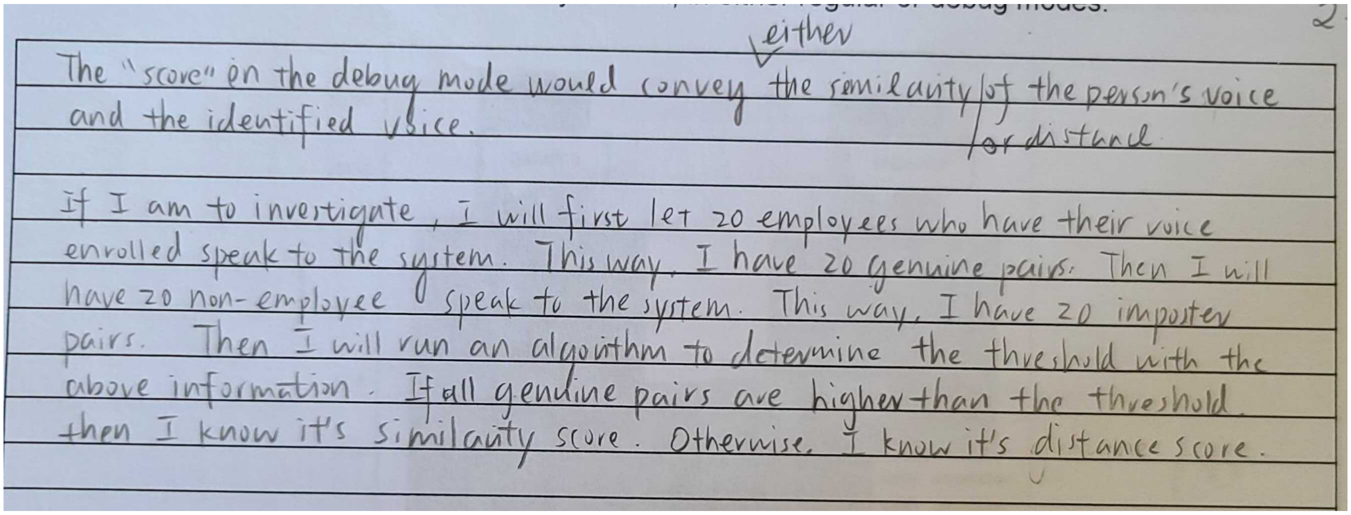
[Question 1] (2 points)
Suppose you were hired by a company to coordinate the deployment of an access management system to control the entrance of authorized employees to their “special operations building”. Among many off-the-shelf available solutions, you found a speaker recognition system whose software interface documentation is rather succinct, with only three exposed functions:

Function	Name	Input	Output (regular operation)	Output (debug mode)
1	enroll	(1) Audio file path: string	(1) Success: boolean	(1) Success: boolean, (2) Id: integer
2	identify	(1) Audio file path: string	(1) Success: boolean	(1) Success: boolean, (2) Id: integer, (3) **Score**: float
3	verify	(1) Audio file path: string, (2) Id: integer	(1) Success: boolean	(1) Success: boolean, (2) **Score**: float

The three functions can operate in two distinct modes (either in “regular” or in “debug” mode), with each one leading to different output behaviors. While the fourth column of the table above details the output in a regular system operation, the last column details the system output in “debug” (or diagnostic) mode. The figure below depicts the hardware interface of the system when operating in debug mode. As one might observe, there is an embedded digital display (represented by a black rectangle, besides the circular microphone) that freely shows the function outputs.



Without further information and based on your experience with biometric systems, what would the "Score" outputs in debug mode convey? If you were to investigate and establish their meaning (e.g., distance, similarity, confidence, etc.), how would you proceed? Please describe it in detail. Consider that you have the provided software fully operational and, therefore, you are able and free to enroll, identify, and verify as many individuals as you want, in either regular or debug modes.

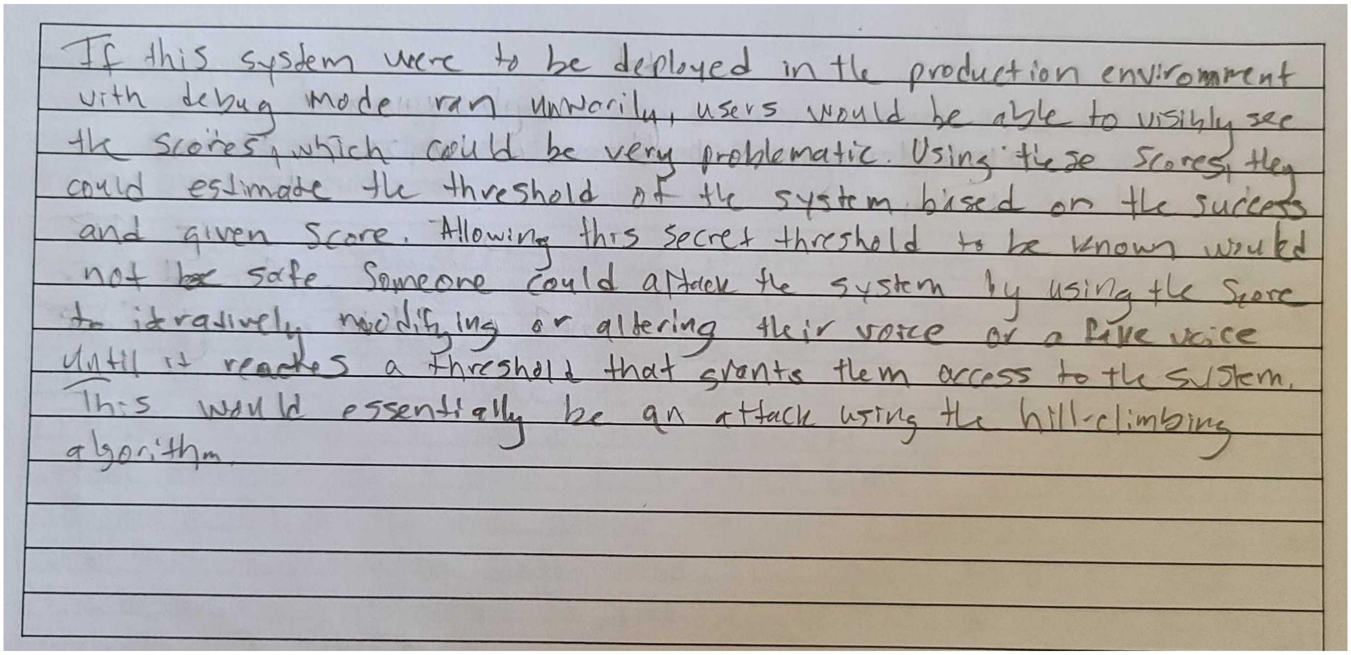


2

The "score" on the debug mode would convey either the similarity or distance of the person's voice and the identified voice.

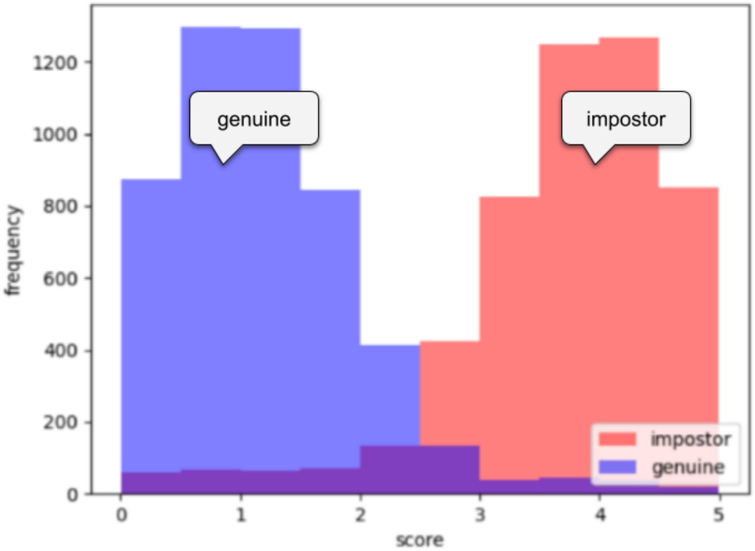
If I am to invertigate, I will first let 20 employees who have their voice enrolled speak to the system. This way, I have 20 genuine pairs. Then I will have 20 non-employee speak to the system. This way, I have 20 imposter pairs. Then I will run an algorithm to determine the threshold with the above information. If all genuine pairs are higher than the threshold then I know it's similarity score. Otherwise, I know it's distance score.

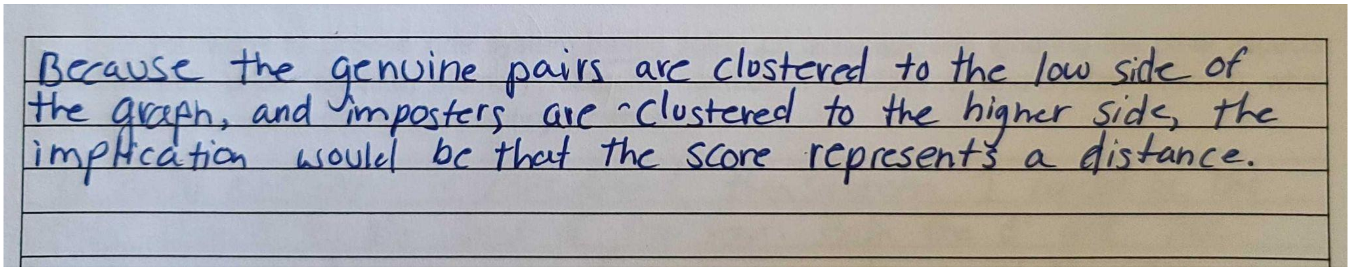
[Question 2] (2 points)
How problematic would it be to deploy this speaker recognition system in the production environment and let it run unwarily in debug mode? If someone were to exploit these exposed scores, how could they attack the system? Please explain in detail.



If this system were to be deployed in the production enviromment with debug mode ran unwarily, users would be able to visibly see the scores, which could be very problematic. Using these scores, they could estimate the threshold of the system based on the success and given score. Allowing this secret threshold to be known would not ~~be~~ safe Someone could attack the system by using the score to itratively modifying or altering their voice or a fake voice until it reaches a threshold that grants them access to the system. This would essentially be an attack using the hill-climbing algorithm.

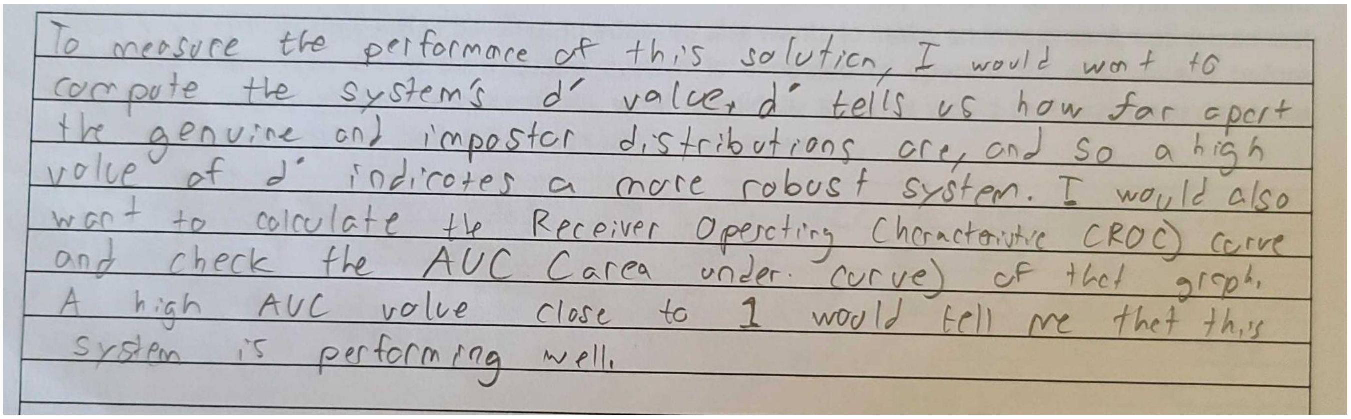
[Question 3] (1 point)
Suppose that after investigating the operation of the speaker recognition system, you have obtained the graph below, with the impostor versus genuine distributions of the system scores for five thousand observations of each group. Based on this graph, does the score convey a distance or a similarity of the voice presentation to the template stored in the system database? Please justify your answer.





Because the genuine pairs are clustered to the low side of the graph, and imposters are clustered to the higher side, the implication would be that the score represents a distance.

[Question 4] (1 point)
Considering the type of the system's score (either similarity or distance), if you were to measure the performance of this solution, how would you proceed? Please describe what metrics you would report and what graphs you would generate.

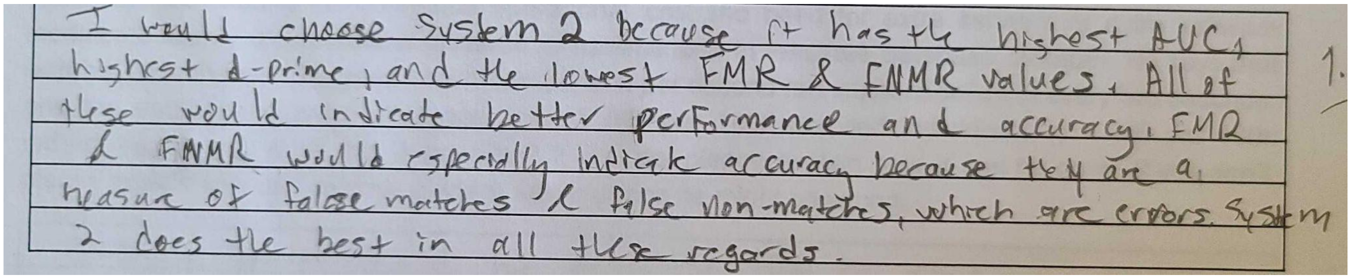


To measure the performance of this solution, I would want to compute the system's d' value, d' tells us how far apart the genuine and impostor distributions are, and so a high value of d' indicates a more robust system. I would also want to calculate the Receiver Operating Characteristic (ROC) curve and check the AUC (area under curve) of that graph. A high AUC value close to 1 would tell me that this system is performing well.

[Question 5] (1 point)
Besides the previously mentioned speaker recognition system, there are three other off-the-shelf well-documented biometric systems available for acquisition. The table below summarizes these solutions after a careful reading of their specs.

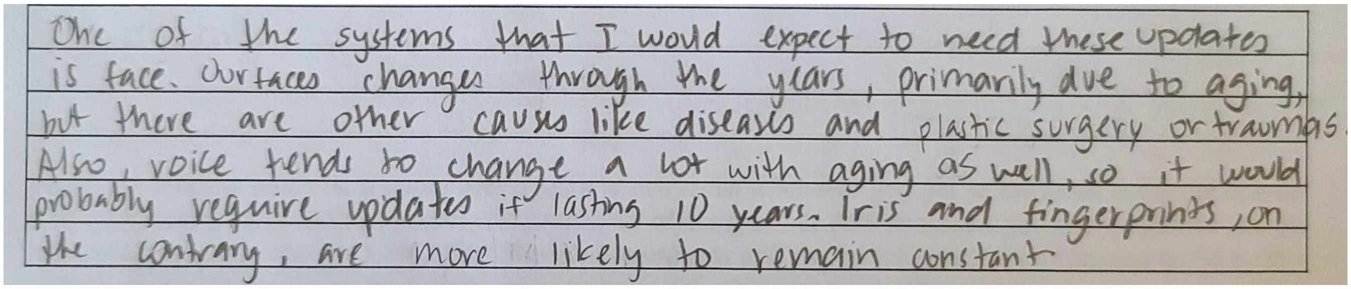
	System 1	System 2	System 3	System 4
Trait	Voice	Face	Fingerprint	Iris
AUC	0.96	0.98	0.97	0.92
d-prime	2.94	4.09	2.80	2.35
FMR @ EER	0.0675	0.0027	0.0554	0.0912
FNMR @ EER	0.0675	0.0027	0.0554	0.0912
Price	$25,000.00	$10,000.00	$2,500.00	$5,000.00
Runtime (comparisons per sec.)	2,500	1,000	1	100
Database storage (MB per 100k individuals)	160	200	2	780

If you were to choose one system based solely on accuracy and ignoring the other aspects (such as trait, price, runtime, memory footprint, number of employees, and system lifetime), what solution would you select? Please justify your answer.



I would choose System 2 because it has the highest AUC, highest d-prime, and the lowest FMR & FNMR values. All of these would indicate better performance and accuracy. FMR & FNMR would especially indicak accuracy because they are a measure of false matches & false non-matches, which are errors. System 2 does the best in all these regards. 1.

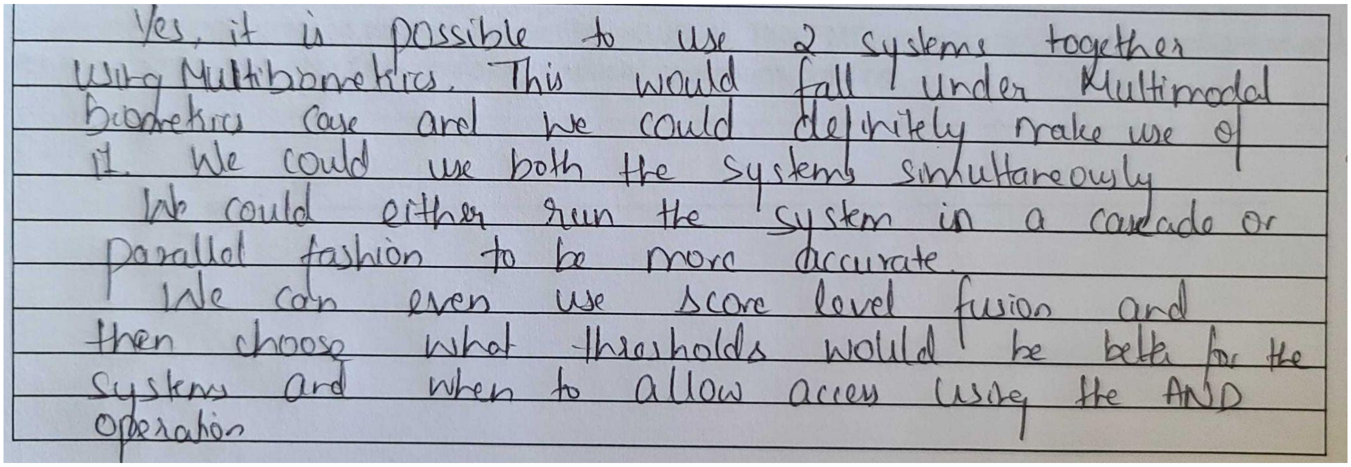
[Question 6] (1 point)
Your boss just brought a little bit more information to the table. Only around 50 employees will need access to the special operations building. In addition, she wants to make an investment that should last at least 10 years (i.e., the to-be-acquired system is expected to operate for one decade before replacement). Based on these requirements, what candidate systems would probably need database template updates along their lifetime? Please justify your answer.



One of the systems that I would expect to need these updates is face. Our faces changes through the years, primarily due to aging, but there are other causes like diseases and plastic surgery or traumas. Also, voice tends to change a lot with aging as well, so it would probably require updates if lasting 10 years. Iris and fingerprints, on the contrary, are more likely to remain constant

[Question 7] (2 points)

Your boss was intrigued that the two cheapest systems altogether cost less than the other solutions. She was wondering if there is any advantage to acquiring these two systems instead of a single and more expensive one. Does she have a good point? What would be the possibilities if the company acquires the two cheapest solutions? Would you be able to leverage them both? If you would, please explain how you could do it.



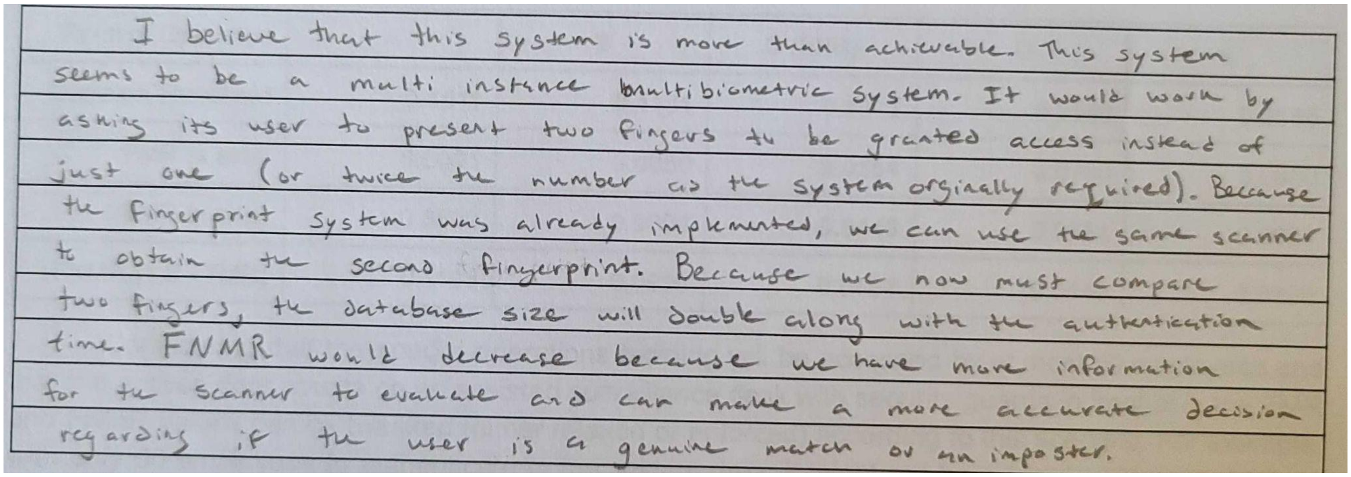
Yes, it is possible to use 2 systems together Using Multibiometrics. This would fall under Multimodal biometrics case and we could definitely make use of it. We could use both the systems simultaneously

We could either run the system in a cascade or parallel fashion to be more accurate.

We can even use score level fusion and then choose what thresholds would be better for the systems and when to allow access using the AND operation

[Question 8] (2 points)

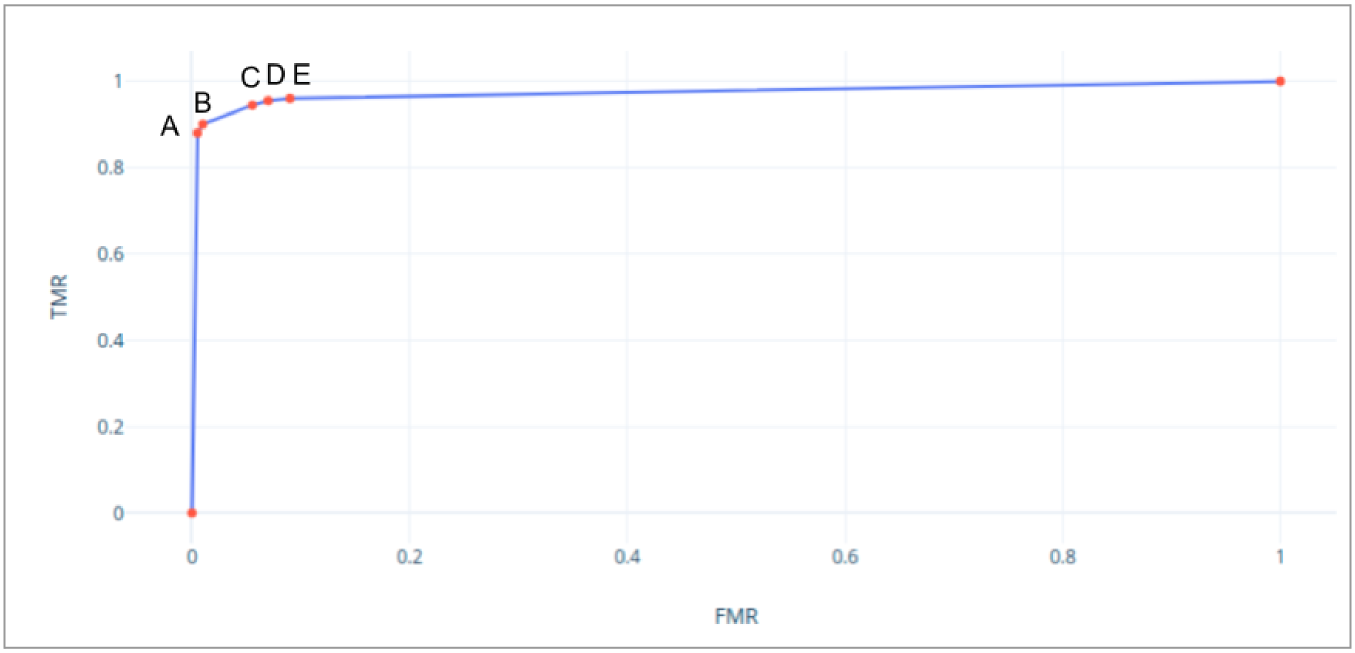
One of the software engineers of the company came to you claiming that he knows how to leverage the cheapest fingerprint-based solution alone in a way that improves its FNMR from 0.0554 (5.54% of probability of false rejection) to 0.0554 x 0.0554 = 0.00306916 (0.306916% of probability of false rejection), with nearly no additional operational cost (no need for extra sensors or extra software modules but only an affordable increase in runtime and in template database storage). He says that with his idea, the runtime goes from 1 comparison per second to 1 authentication in every two seconds, and the database storage increases from 2 MB per 100k enrolled individuals to 4 MB per 100k individuals. Do you think this is possible? If you do, please explain how it can be done. If you don't, please explain why and regarding which assumptions he might be wrong.



I believe that this system is more than achievable. This system seems to be a multi instance multibiometric system. It would work by asking its user to present two fingers to be granted access instead of just one (or twice the number as the system orginally required). Because the fingerprint system was already implemented, we can use the same scanner to obtain the second fingerprint. Because we now must compare two fingers, the database size will double along with the authentication time. FNMR would decrease because we have more information for the scanner to evaluate and can make a more accurate decision regarding if the user is a genuine match or an imposter.

Useful tip. Imagine you're in a game with dice and you lose whenever you get all the dice facing the one-dot side after tossing them. With one dice, your probability of losing is 1/6. With two dice, you lose only when you get one dice facing one dot AND the other dice also facing one dot; hence the probability of losing is 1/6 x 1/6 = 1/36. Analogously, in a biometric system, whenever you get only false rejections, you lose.

[Question 9] (2 points)
The graph below depicts the ROC curve of the cheapest solution (fingerprint-based), whose AUC is 0.97. Within this graph, 5 interesting points of operation are highlighted, from A to E. Point C corresponds to the system operation at an equal error rate (EER), when FNMR = FMR = 0.0554, and the decision score threshold is set to 0.3212. In this configuration, the system wrongly rejects nearly 5% of genuine authentications (i.e., one in every twenty genuine users is wrongly denied access, hence FNMR=0.0554), and wrongly accepts 5% of impostor authentications (i.e., one in every twenty impostor users are wrongly granted access, hence FMR=0.0554). This FMR, in particular, is not acceptable at all for ensuring the security of the company's special operations building.

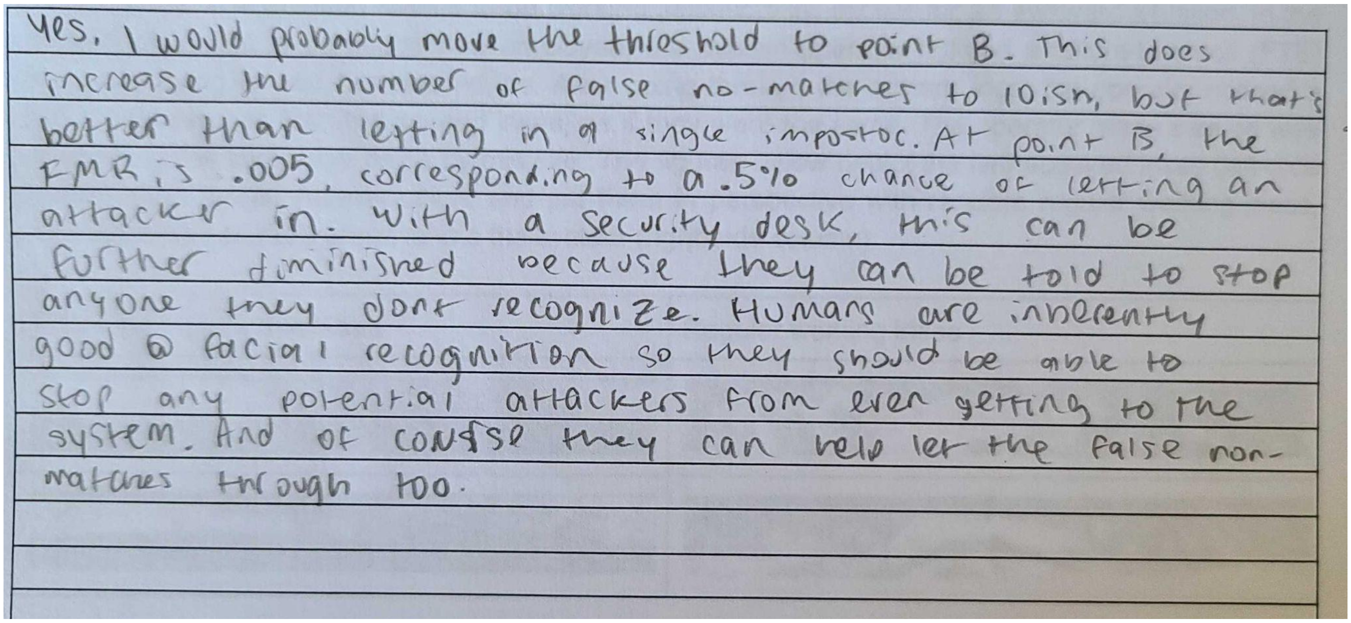


The table below details each one of these 5 points of operation, in terms of decision score **threshold**, **FMR**, **FNMR**, and **TMR**.

Point of Operation	A	B	**C (EER)**	D	E
Decision Threshold	0.4511	0.4131	**0.3212**	0.2956	0.2585
FMR (x-axis)	0.0001	0.0050	**0.0554**	0.0700	0.0900
TMR (y-axis)	0.8802	0.9004	**0.9446**	0.9551	0.9601
FNMR (1.0 - TMR)	0.1198	0.0996	**0.0554**	0.0449	0.0399

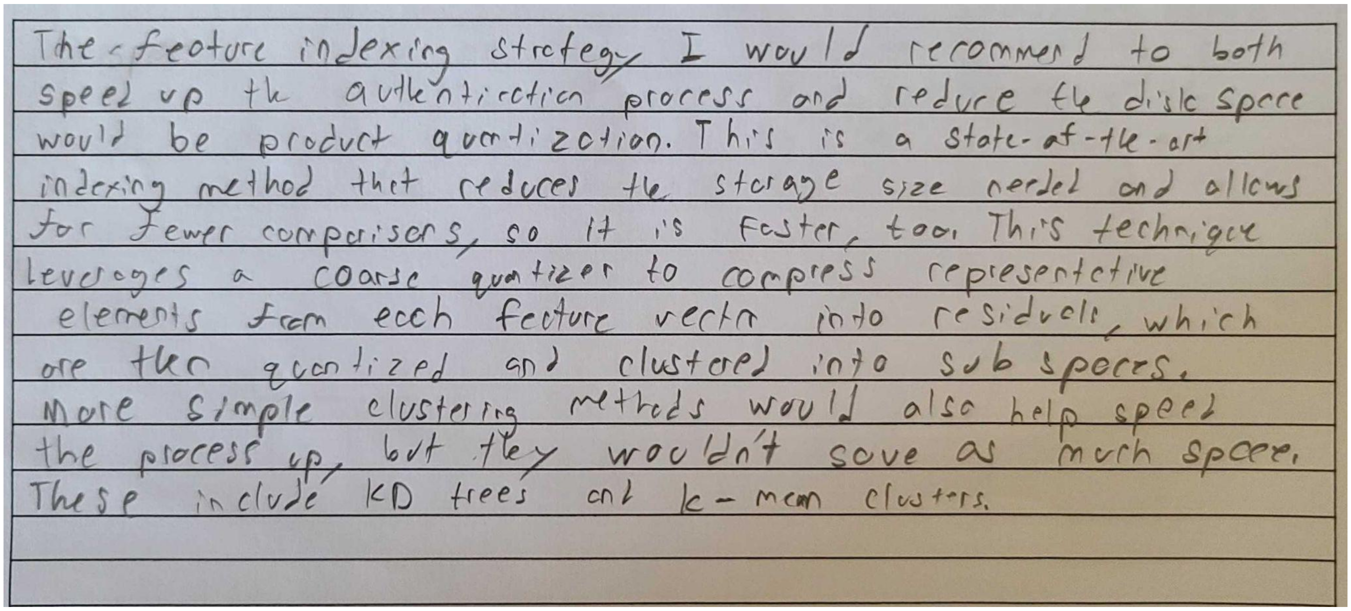
Considering that the special operations building will be accessed by at most 50 employees and that the access door counts on an assisted surveillance desk with security guards in front of it, the FMR and FNMR values can be tweaked (either relaxed or enforced) according to this scenario. For example, with only 50 employees to authenticate in the system daily, it might not be a big deal to wrongly deny access to five of them every day (i.e., tolerate an FNMR of 10%), considering that the guards will be there to supervise these situations and manually let the five wrongly denied folks in. In this case, a low FMR is much more important than a low FNMR, indicating that the operation at EER might not be the best choice.

Given these considerations, is there a way to still use the cheapest fingerprint-based biometric system with no fusion at all? If you were to do it, how would you proceed? Please justify your answer.



Yes. I would probably move the threshold to point B. This does increase the number of false no-matches to 10ish, but that's better than letting in a single impostor. At point B, the FMR is .005, corresponding to a .5% chance of letting an attacker in. With a security desk, this can be further diminished because they can be told to stop anyone they dont recognize. Humans are inherently good @ facial recognition so they should be able to stop any potential attackers from ever getting to the system. And of course they can help let the false non-matches through too.

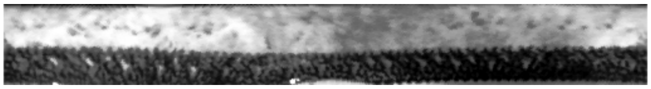
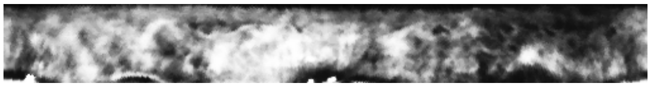
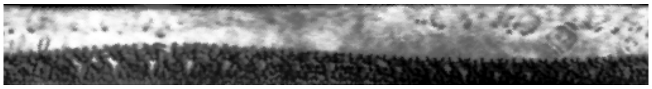
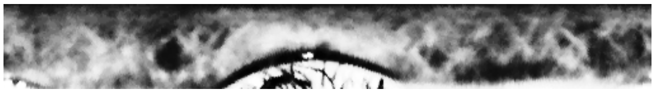
[Question 10] (2 points)
After some debate and because of a better offered lifetime support by the respective manufacturer, your company decided to acquire the iris recognition system, to authenticate both eyes of the employees. This system thus spends 2 x 780 = 1560 MB of disk for every 100k enrolled individuals (given both eyes are enrolled). If you were to use this system in a large scale scenario, for authenticating millions of individuals, what feature indexing strategies would you use to (1) speed up the authentication process and reduce the system runtime, and (2) reduce the disk space spent to store the enrolled iris templates. Please justify your choices of feature indexing method.



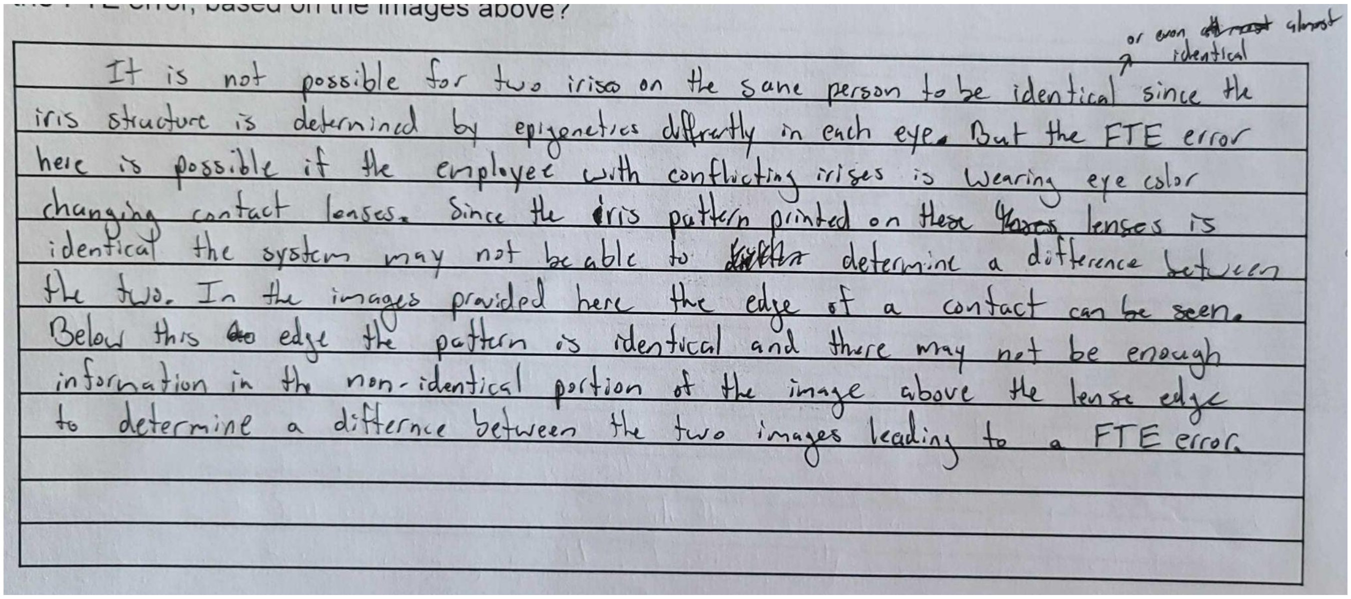
The feature indexing strategy I would recommend to both speed up the authentication process and reduce the disk space would be product quantization. This is a state-of-the-art indexing method that reduces the storage size needed and allows for fewer comparisons, so it is faster, too. This technique leverages a coarse quantizer to compress representative elements from each feature vector into residuals, which are then quantized and clustered into sub spaces. More simple clustering methods would also help speed the process up, but they wouldn't save as much space. These include KD trees and k-mean clusters.

[Question 11] (2 points)

Congratulations! After your guidance, the iris recognition system was acquired, set up, and deployed on your company. The solution seems to be working satisfactorily, except for an awkward situation. In the occasion of adding a particular new employee, the system operator noticed a failure-to-enroll (FTE) error while trying to add her second iris. After going through the system logs, the operator noticed a conflict between her first and second irises, as if they were the same. The operator made sure he was not enrolling the same eye twice by mistake. The figures below depict the two acquired irises (left-side column), after proper normalization, and put them in perspective with random regular working irises, which were successfully enrolled into the system (right-side column).

Employee's conflicting irises	Regular working irises
	
	

Based on your experience, is it possible for someone to have left and right irises nearly the same? Please justify your answer. In the case you claim it is not possible, what could be the reason for the FTE error, based on the images above?

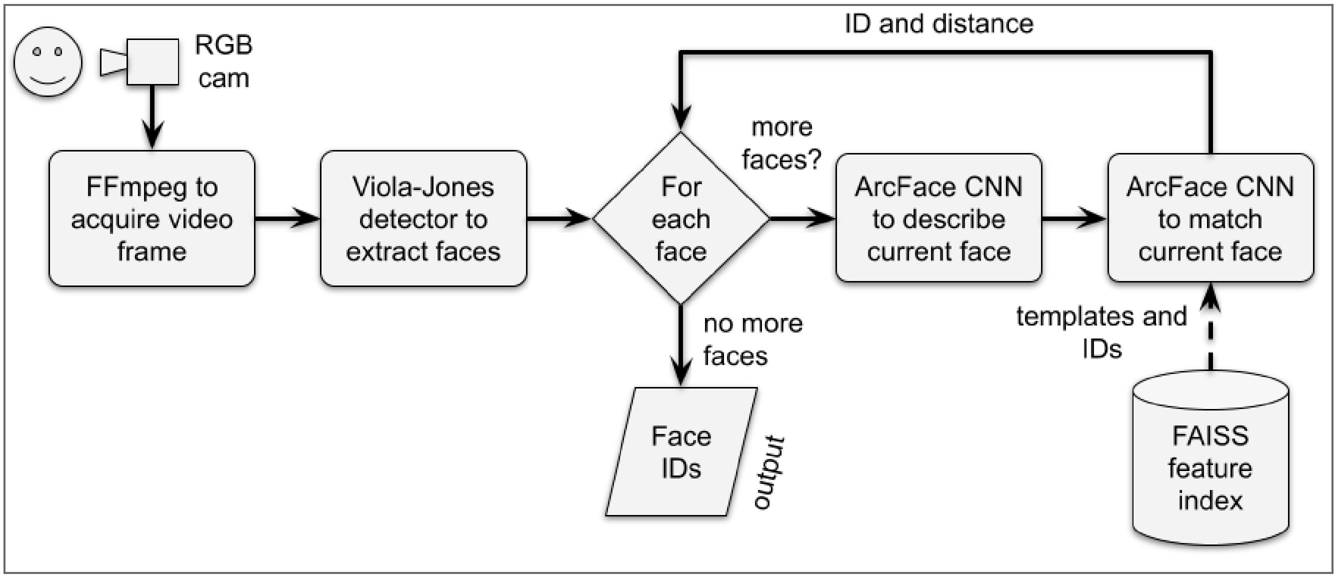


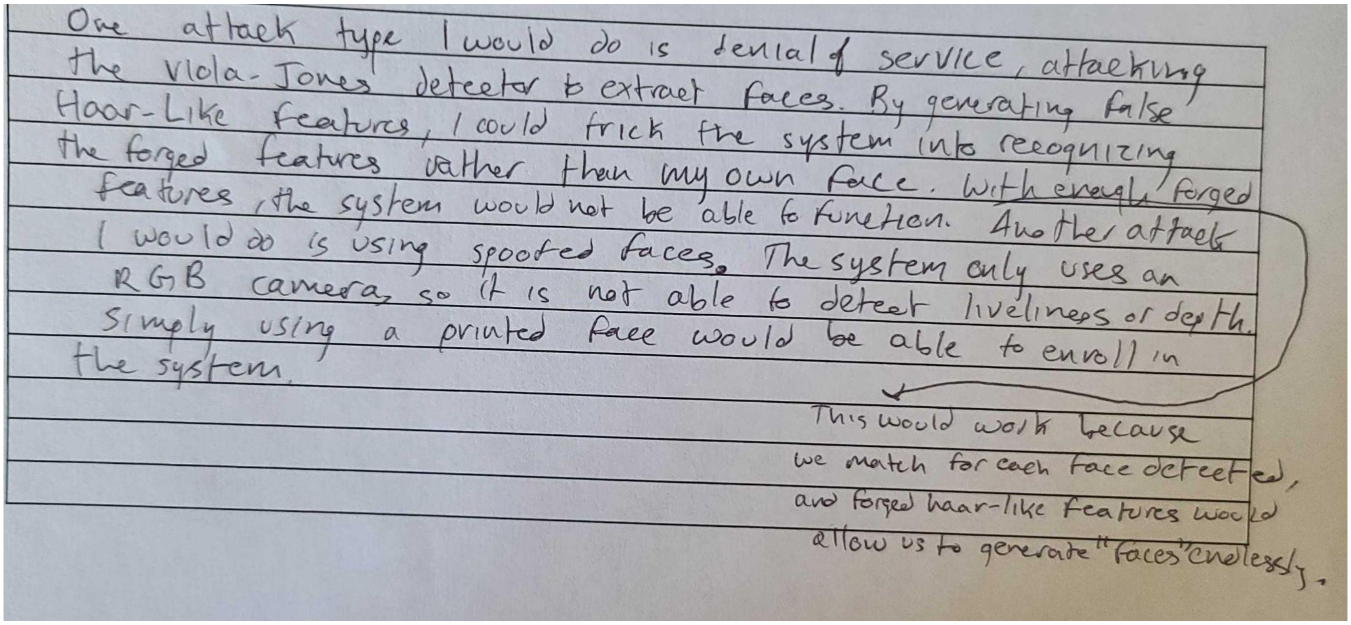
the FTE error, based on the images above?

It is not possible for two irises on the same person to be identical or even ~~almost~~ almost identical since the iris structure is determined by epigenetics diffrently in each eye. But the FTE error here is possible if the employee with conflicting irises is wearing eye color changing contact lenses. Since the iris pattern printed on these ~~[illegible]~~ lenses is identical the system may not be able to ~~[illegible]~~ determine a difference between the two. In the images provided here the edge of a contact can be seen. Below this ~~do~~ edge the pattern is identical and there may not be enough information in the non-identical portion of the image above the lense edge to determine a differnce between the two images leading to a FTE error.

[Question 12] (2 points)

The diagram below details the implementation modules of the discarded face recognition solution. If you were to perform two different types of white-box attacks on this system (such as repudiation, spoofing, or denial of service), what would they be? Please explain your answer.


RGB cam
FFmpeg to acquire video frame
Viola-Jones detector to extract faces
For each face
more faces?
ArcFace CNN to describe current face
ArcFace CNN to match current face
ID and distance
no more faces
Face IDs
output
templates and IDs
FAISS feature index



One attack type I would do is denial of service, attacking the Viola-Jones detector to extract faces. By generating false Haar-Like features, I could trick the system into recognizing the forged features rather than my own face. With enough forged features, the system would not be able to function. Another attack I would do is using spoofed faces. The system only uses an RGB camera, so it is not able to detect liveliness or depth. Simply using a printed face would be able to enroll in the system.

This would work because we match for each face detected, and forged haar-like features would allow us to generate "faces" endlessly.