Final Assignment: Identity Verification

By Anthony Ayala, MSBA Class of 2024

Due Date: December 12th, 2023

Table of Contents:

Case One: Report on Facial Comparisons with Face Recognition vs AWS Rekognition

The use of Facial Biometrics allows to reliably verify the identities by comparing faces, and the methods I used was the open-source package called Face Recognition and my work is compare the performance, the pros, and cons of this package vs Amazon's Rekognition API service. With both methods performed in this case, I will go over my key findings, highlight the confusion matrix each method produced, make comparisons to the model metrics (accuracy, precision, and recall), and highlight the differences between the two methods on how they manage no matches and generate similarity scores.

Beginning with key findings, AWS produces the highest match similarity scores on average and split on prediction as it generated 89 matches and 87 no-matches with a similarity threshold of 80. The reason for choosing a high threshold is matter of optimism of AWS's service as it is regarded as one of the most important players in the cloud computing industry. As for facial recognition, it produces matching similarity scores ranging from 0 to 70 and most comparisons fall anywhere between 50 and 60 and predicted 48 more matches than no-matches with a similarity threshold of 50. Though AWS's Rekognition had a higher threshold, it is apparent that this method will produce highly confident scores and matches and will not accept matches under the threshold. While for facial recognition the results scores might not be as high, it can give us perspective on how models could use some improvements as the majority would find that aged photos to be remarkably similar and not just 50%.

AWS correctly predicted 89 matches and 50 no-matches but predicted 37 no matches when there was an expected match and predicted 0 matches when there was no match. These numbers define AWS's model metrics to have an accuracy of 78.97%, a precision of 100% and recall of 70.63%. While face recognition correctly predicted 100 matches and 48 no-matches but predicted 16 no-matches when there was an expected match and predicted a man when there was no expected match. These numbers show that facial recognition model metrics to have an accuracy of 89.77%, a precision of 98.21% and a recall of 87.30%. The AWS model does better in making the most correct predictions when predicting a match, but the AWS model falls short as facial recognition does better at making more correct predictions out all predictions and making more correct predictions when there is an expected match.

For highlighting the differences between the methods, I have found Amazon's Rekognition API service to be superior regarding performance, ease of running code, data storage, and generating similarity scores with high confidence. However, it is not 100% perfect as when it does work properly it does a phenomenal job but when threshold is set too high (80), then it happens to be that this method needs some adjusting. For example, table 1 shows match similarity score and predicted math was generated from AWS and looking specifically at row 2 with Qian Chen, her expected match was said to be a "match," but AWS gave a score 0.00 and predicted no match. While in table 2 which used facial recognition, Qian Chen's match similarity as 55.156 and predicted to "match." As for handling non-matches, face recognition is

more dependable than AWS as we can see Qian Chen's photo in image one where she does match her aged photo.

Overall, I have found AWS to be more trustworthy and easier to experiment with as matches can be done much faster than generating encodings and using those encodings to then compare faces with face recognition. With AWS, I could easily upload a later semester image of myself and have AWS generate the similarity score of my headshot and later image of myself! I also like that AWS's detect face functions go beyond by detecting if a person has glasses, no beard and extra details that facial recognition is limited to detecting a mouth, face, eyes and a nose. At the end of the day, both models work well but I do find AWS to be the better option to go with but keeping in mind of threshold will truly determine the predictions so one must be mindful of that.

Table 1: AWS Rekognition Match Similarity Scores and Predicting Matches

	match_id	source_image	name	target_image	expected_match	match_similarity	predicted_match
0	100	Bingyu_Li.jpg	Bingyu_Li	Bingyu_Li_aged.jpg	match	62.974	match
1	101	Whitney_Joyce_Isbell.jpg	Whitney_Joyce_Isbell	Whitney_Joyce_Isbell_aged.jpg	match	56.220	match
2	102	Qian_Chen.jpg	Qian_Chen	Qian_Chen_aged.jpg	match	55.915	match
3	103	Hanshuai_Shi.jpg	Hanshuai_Shi	Hanshuai_Shi_aged.jpg	match	59.156	match
4	104	Ruochen_Bao.jpg	Ruochen_Bao	Ruochen_Bao_aged.jpg	match	29.604	no-match

Table 2: Face Recognition Match Similarity Scores and Predicting Matches

	match_id	source_image	name	target_image	expected_match	match_similarity	predicted_match
0	100	Bingyu_Li.jpg	Bingyu_Li	Bingyu_Li_aged.jpg	match	98.274	match
1	101	Whitney_Joyce_Isbell.jpg	Whitney_Joyce_Isbell	Whitney_Joyce_Isbell_aged.jpg	match	97.822	match
2	102	Qian_Chen.jpg	Qian_Chen	Qian_Chen_aged.jpg	match	0.000	no-match
3	103	Hanshuai_Shi.jpg	Hanshuai_Shi	Hanshuai_Shi_aged.jpg	match	89.793	match
4	104	Ruochen_Bao.jpg	Ruochen_Bao	Ruochen_Bao_aged.jpg	match	0.000	no-match

Image One: Face Recognition Match Similarity for Qian Chen

Report on Who Came to Class with Face Recognition vs AWS Rekognition

The use of Facial Biometrics allows to reliably verify the identities of my classmates in my cohort who showed up to class, and the methods I used was the open-source package called Face Recognition and my work is compare the performance, the pros and cons of this package vs Amazon's Rekognition API service. I performed identity verification on three images of my cohort and then piled up the results into a single data frame. With both methods performed in this case, I will go over my key findings, making comparisons histogram of the similarity scores, make comparisons and highlight the differences between the two methods on how they manage no matches, generating similarity scores and overall preferences.

Beginning with key findings, in image one down below AWS seems to overcount by matching multiple images from the buckets/collections to a face in a photo as it a total of 34 matched faces. AWS also does a phenomenal job when it comes to generating similarity scores as the majority of the scores are high (between 82.5% and 100%) and AWS has high confidence in these scores. Also, when there is a face that is not in the bucket or collection, then Amazon will be a red box meaning that there is no match. As for facial recognition, it does not over count the detecting of matching faces and just gives one match per detected face, which is simple and effective. Face recognition will do its best to match a detected face with a face encoding that is closest to the detected face, meaning that every detected face has a match and there are no non-matches. It also has similar scores that mostly reside around 40% or to 60% which is much lower than AWS. It becomes clear that AWS does better in handling generating similarity scores and handling no detected faces, but facial recognition does a much better job in detecting faces and not overcounting.

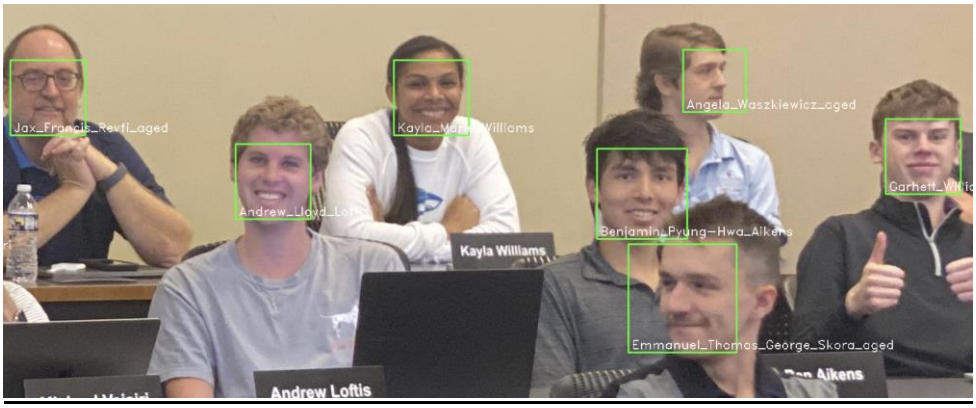
What I have found interesting in the differences between these two methods happens to be the detection of faces and facing error. Starting with the detection of faces, AWS ended up creating duplicate results at times for certain students, it really liked matching aged photos of certain students in the attendance photos, and liked to match another attendance photo that may have the same students in the photo that was analyzed. AWS had a fair bit of cleaning up data and these unique results were taken into consideration when making the final comparison. As for facial recognition, it did not have any of these issues but rather it tried to match our friends from IBM to students who may look like them which raises a lot of bias. The one instance where facial recognition had a slight error was in image two where this method did not correctly match Cole's detected face as facial recognition believed Cole looked like Angela's aged photo. AWS was able to detect Cole correctly and our friends in IBM were highlighted with red boxes meaning they were not part of the S3 buckets or collections and hence were not included in the data frame.

Overall, I have found AWS again to be more trustworthy and easier to experiment with as matches can be done much faster than generating encodings and using those encodings to then compare faces with face recognition. However, AWS this time called for more data clean up and could have used some filtering in handling aged photos which I believe is something that is

a relatively easy fix! At the end of the day, both models work well, and I think it really comes down to whether the company prefers to have higher matching similarity scores or simply detecting faces. If it comes down to accuracy then I believe AWS is the best, but if it comes down to simply detecting the count of faces then I would say facial recognition will do better.

Image One: AWS Rekognition's Issue of Matching too many faces

Image Two: Facial Recognition creating bias and not detecting Cole's face correctly.



Executive Summary

After performing extensive analysis on Facial Biometrics with Face Recognition and AWS Rekognition my summary covers the specific details of both models but before we dive into the nitty gritty details, I must first cover the methodology and practices employed in this work.

The project's goal is verify who shows up to class and be confident in comparing faces between a student and their aged photo. In case one, the methodology involves taking headshot photos of all the students, comparing each student with their aged photo, and generating a similarity score. This enables us to predict matches or no-matches for each student and their aged photo. In other words, we were given data that states each student and their aged photo have an expected match and with both models we would like to see which model does a better job and correctly predict a match and a no-match. In case two, the methodology involving matching students' headshot photos and to a photo of the students who attended class, with a match is determined by a threshold. In simpler terms, case two's work allows us to see who truly did show up to class and who did not show up to class.

In the key findings for case one, AWS Rekognition produces high similarity scores with high confidence, performing the best in making correct predictions when predicting a match. However, facial recognition excels in accuracy as it made the most correct predictions out of all the predictions and does better than AWS at making correct predictions when there is an expected match. What I recognize as AWS best attributes are its ease of running and understanding code, generating realistically accurate similarity scores, and its effectiveness as it does reduce time by about 50% when it comes to generating encodings and data which facial recognition took roughly 20 minutes to get encodings. Although this is true, I see that facial recognition's best attributes are its reliability when everything works correctly especially in the case for matching Qian's Chen photo to her aged photo which AWS predicted there was no match and gave a score of 0.0% mentioned in the report for case one, fourth paragraph. Speaking of facial recognition's capabilities, I see it falling short when making comparisons as at times it compares one student to another student and not compare a student and their aged photo. Beyond that, AWS is slightly more reliable, and its ease of analysis will be more apparent in the next few paragraphs.

AWS might have lower accuracy and lower recall in comparison to face recognition's numbers, but the model is 100% precise. These numbers are still extraordinarily strong when determining if a model is good or not, and the match similarity for a student and their aged photo had a high threshold which meant AWS predict matches when the compared faces have a match similarity score of 80% or greater. While for face recognition, the hope was to just to predict a match when the match similarity score is greater than 50%. AWS still shows lots of promise as this model can give realistically accurate scores of matches similarities when it comes to compare a student's face to their aged photo as seen in image one in the appendix down below. Also, AWS shows its strength in identifying a student that showed up in class like in image two in the appendix as Ruiqi Cheng, the face not detected, was highlighted in a red box to signify that there was no match for her since she simply did not a have headshot photo. This does an excellent job in comparison to face recognition as face recognition creates bias by identifying Ruiqi Cheng to be Jiacheng Wang as seen in image three in the appendix when facial recognition should detect no non-matches or when a face is not the encodings. Finally, when AWS detected the students in the second photo it was noticeably confident in identifying the students as the similarity scores were all greater than 95%.

We understand that there is a clear winner in terms of the model performance and its effectiveness, but AWS does face quite a few obstacles. One of them being the matching two faces to a single detected photo like in the image two in the appendix where there are clearly nine students in the photo and AWS counted twenty-one matches and one no-match. The complication here lies within the buckets and collection of images, which AWS matched the students who showed up to class with not only their headshot but also their aged photo as well and another attendance photo which could have included the same students. What this causes is extra time in cleaning up the data and removing duplicates along with students who may have been matched with their age photo. The only other issue that may call for mindfulness is setting the threshold, which needs an explanation why a particular number was chosen but AWS at times did not generate high match similarity scores for students who do very much look their aged photo, referring to the case of Qian Chen. These two instances were key issues for AWS, but overall are simple fixes. The preference does not change as AWS still reigns as being the superior model as its problems are not as big as they may sound to be, and its reliability shows when identifying who showed up to class and when comparing faces.

When considering the case that if face recognition was on the cloud and if its precision was 100%, AWS is preferred. Among the many reasons for that belief lies within the maintenance, effectiveness, ease, efficiency. For maintenance and efficiency, I do not see a better competitor when it comes to storing data into the buckets and uploading images to a collection and instantly producing results for comparing faces and identifying who showed up to class. The analysis is completely streamlined and shaves a lot of time while face recognition would spend most of its time generating encodings and taking longer to produce the results that AWS would do in less time. In terms of effectiveness, AWS still stands strong by producing great model metrics and generating high similarity and match similarity scores with a high threshold, which again demonstrates its trustworthiness and reliability. When it comes to ease, AWS has allowed me to compare a later semester image of myself by simply uploading the photo to the bucket and not have to generate an encoding then compare that encoding to my headshot photo's encodings like facial recognition requires. Beyond that this model is not perfect as does this model comes at a price, while face recognition does not but when it comes to paying a price that is very much fractional as AWS S3 Standard charges the first 50 TB per month for $0.023 per GB, and it saves so much time and isn't complex like face recognition by dealing with arrays. Then by all means AWS still reigns as the best model for both case one and case two by not being super complex and is super-efficient yet also effective.

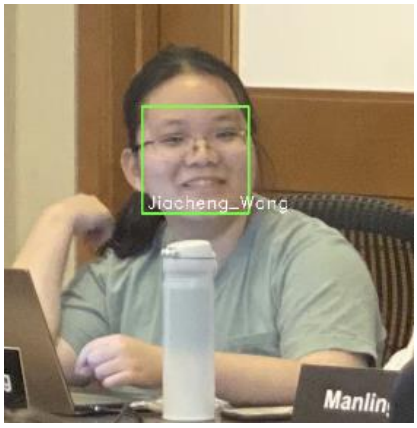
My recommendations are to use AWS when it comes to both cases for the reasons mentioned previously, which highlight impressive performance, there are more pros than cons, and the issues that AWS faced were simple fixes. What I also would share is that when it comes to predicting matches, I would decrease the threshold for AWS especially in handling the case for Qian Chen does look much like her aged photo although AWS may have yielded a match similarity score lower than 80%, which ended up predicting there was no match there. Finally, when it comes to comparing faces and verifying the identities of the students who showed up to class AWS makes it simple for everybody to understand believe when looking at the analysis at first glance, especially when no matches are highlighted in a red box and students and their aged photos should very much look similar.

Appendix:

1. AWS Compare Faces, High Match Similarity Score of 93.73%

2. **AWS Detecting a No-Match in the Second Attendance Photo (One No Match, Ruiqi Cheng)**

3. **Face Recognition Faces Bias**



4. AWS Names of Faces Detected and Similarity Scores of the Second Attendance Photo

	ExternalImageId	Similarity
16	Xiaoyu_Zong	95.780121
17	Yutong_Ouyang	99.023392
14	Tianyu_Cui	99.595695
7	Kaushik_Rajaram	99.707680
12	Manling_Shi	99.930420
2	Meghan_O_Malley	99.982353
9	Michelle_Monica_Saikali	99.990616
5	Amanda_Renner_Gild	99.995033