Performance Comparison of Free vs. Commercial Face Detection and Landmark Localization Systems for Oblique and Profile Views, I love you

Author: Jinlong Lin
Supervisor: Kobus Barnard, Hong Cui

1 Abstract

In facial recognition, landmarks are specific and meaningful locations on the face, such as the outer corner of an eye, as illustrated in Figure 1. It is common and essential to find such coordinates in images for facial recognition and analysis software. The goal of this project is to understand how three facial recognition and analysis programs, open-source OpenFace (Baltrusaitis, Robinson & Morency,2016), Deva (Zhu & Ramanan, 2012), and closed-source Face++ (Beijing Kuangshi Technology Company, 2019), work in the wild and their performance under different imaging conditions, exploring the causes of performance differences. Their performances are compared on finding faces and landmark locations in the 300w dataset (Sagonas et al, 2016) and AFLW dataset (Koestinger et al, 2011), and their predictions are analyzed both qualitatively and quantitatively through manual and automated processes. The analysis involves visualizing landmarks on top of images, computing the confusion matrix of face detection decisions, and computing the face-size-normalized point-to-point error between a set of predicted and annotated landmark locations. The scripts and the designed analysis methods developed in this project can also be used in the future when the facial recognition programs are updated. The Face++ program has the best F1 score from the confusion matrix in both datasets, and all of three programs perform better on the 300w dataset than the AFLW dataset. Some programs have rather poor landmark location detection for oblique and profile face views.

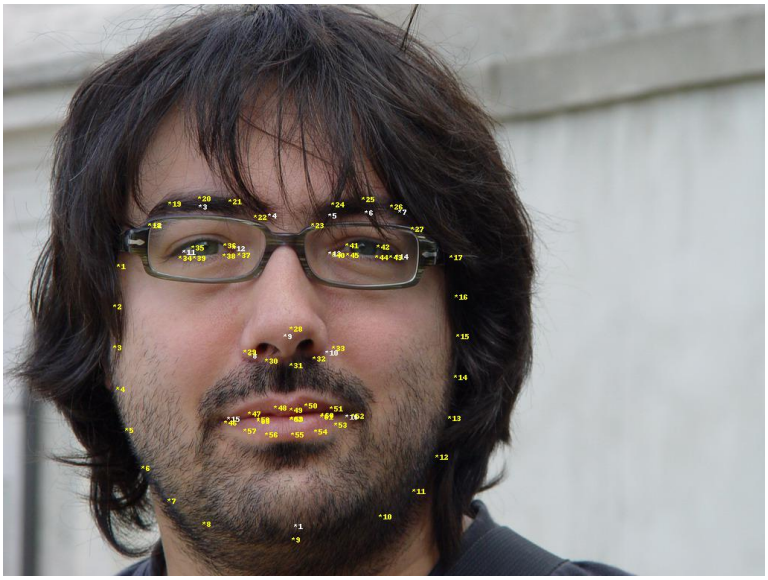

Figure 1. Sample Image Showing Human Annotated (White Points) and Machine Detected Landmarks Coordinates (Yellow Points).

2 Introduction

Facial recognition is a research area where computer algorithms are developed to recognize and extract particular attributes and features of face based on the landmark locations. These facial recognition results can be further used to infer facial expressions and facial activities such as blink rate. The research reported in this report was conducted in the Interdisciplinary Visual Intelligence (IVI)-Labs led by Professor Kobus Barnard in UA Computer Science Department. This Lab seeks to develop methodologies and test and build software that can accurately recognize faces from imagery and video data. Our research group has been developing automated tools for analyzing the output of facial recognition software and compare their performance in specific tasks under different imaging conditions. The most promising three facial recognition tools (OpenFace, Deva, and Face++) were selected and studied in this Capstone Project. A thorough analysis of these tools provides a holistic view of their performance.

This report examines the impact of oblique and profile views face on facial detection. The challenges of this project involve finding a fair and consistent method to compare these facial recognition tools. The methods used are a confusion matrix and a mean error of Euclidean distance. After the comparison, it is found that the Face++ has the best F1 score in both datasets tested (0.98 in 300w dataset, and 0.85 in AFLW). Deva performs better than OpenFace in AFLW dataset, while OpenFace is better than Deva in 300w dataset. In the accuracy analysis of correctly detected faces, OpenFace has a smaller median normalized error than Face++ in 300w dataset, while Face++ is better in AFLW dataset. Deva has larger error than Face++ and OpenFace in both datasets.

3 Datasets and Software Packages

Software packages

-OpenFace (version 2017.2): OpenFace is "an open-source state-of-the art tool intended for facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation." (Baltrusaitis, Robinson & Morency,2016)

-Face++ (version 3.0): Face++ is a commercial company that provides the "leading, reliable" computer vision technology in APIs and SDKs for facial recognition services.(Beijing Kuangshi Technology Company, 2019) Face++ provides a face landmarks API, which locates and returns "key points of face components, including face contour, eye, eyebrow, lip and nose contour etc." (Beijing Kuangshi Technology Company, 2019)

-Deva (version 1.0): Deva is "a unified model for face detection, pose estimation, and landmark estimation in real-world, cluttered images" (Zhu & Ramanan ,2012) created by Carnegie Mellon University. The model is "based on a mixture of trees with a shared pool of parts" (Zhu & Ramanan ,2012).

Datasets:

The AFLW and 300w datasets of annotated face images were selected by our group. AFLW is more challenging than 300w in its imaging conditions. Images in 300w seem to be easier because its cropped images have one face in each image, making the faces in images look more regular. In AFLW, the images were taken from a variety of imaging conditions and contain numerous

monochromatic images. In addition, there are a lot of faces in AFLW that are rotated more than 45 degrees by the mathematical z axis. 300w images has no images like that. Thus, this cropped 300w dataset and AFLW dataset become contrasting datasets. More information on the datasets are provided below.

-300w: The 300-W is a dataset which has in-the-wild images, that is, the images cover "a large variation of identity, expression, illumination conditions, pose, occlusion and face size" (Sagonas et al, 2016). The images were annotated with the 68-point mark-up scheme developed at CMU. (This scheme is also called MultiPIE (MultiPose Illumination, and Expression) (Sagonas et al, 2013 b), see Figure 5. There is a cropped version of the dataset where the images are cropped to only contain one annotated face. We randomly extract 600 images from this cropped dataset.

-AFLW: The Annotated Facial Landmarks in the Wild dataset is a large-scale, real-world database for facial landmark localization. It consists of nearly 26,000 images of faces in a variety of poses, expressions, ethnicities, ages, and genders, taken from a variety of imaging conditions (Koestinger et al, 2011). The images were annotated with the 21-point mark-up scheme. We randomly extract 600 images from dataset and extract the relevant images annotation information from AFLW database.

Data Cleaning and Conversion:

The annotations used different landmark schemes in each of the datasets. 300w dataset uses the 68-point landmark scheme developed at CMU (it is called MultiPIE). AFLW dataset uses the 21-point landmark scheme.

The facial detection systems also used different landmark schemes. Face++ uses 83-point landmark scheme. OpenFace uses 68-point scheme (MultiPIE). Deva uses a different 68-point landmark scheme from MultiPIE. In addition, the number of landmark coordinates generated by Deva is according to the degree the face rotated. When the face is rotated less than 45 degrees by z axis it uses 68-points scheme, and when more than 45 degrees, 39-point landmark scheme is used.

Therefore, to make the facial recognition results comparable across different systems and datasets, it is necessary to convert the Face++ and Deva detection files into 300w's 68-point MultiPIE, which OpenFace also uses. Similarly, the Face++, Deva and OpenFace should be converted into AFLW's 21-point landmark scheme. Further, in Deva, the detection files using 39-points landmark scheme needs to be converted into its 68-points landmark scheme, with the missing point represented using a missing value (NAN).

In the preparing work, the annotation information was extracted from the AFLW databases. The annotation files already available in 300w were used for this dataset. The extracted annotation files were cleaned to ensure they only have the needed landmarks coordinates without extra information. The similar cleaning work for detection files of programs were conducted.

Next, comes the conversion work. Different schemes can be thought of as partially overlapping sets. Some of the points that are not in the overlapping set of one scheme can be converted to a

landmark point in the other scheme, for example, by taking the average of two points that are represented as one point in the other scheme. For the points are not transferable between schemes (points exist in one scheme but not in the others), they were ignored. In a sense, this conversion was to map one landmark scheme into other landmark scheme by computation and ignore the inappropriate points which are not transferable. The landmark scheme conversion is store in the files labeled as Appendix A.

3 Evaluation Metrics

The two evaluation methods employed are a confusion matrix and the mean error of Euclidean distance. The confusion matrix can be used to determine the correct detection (true positive face match) and false detection (false positive face match) and missed detection (false negative face match). And to quantitatively assess the accuracy of the correct detected faces, the mean error of Euclidean distance between manually annotated landmark coordinates and the detection landmark coordinates were computed.

3.1 **Confusion Matrix**

"A confusion matrix summarizes the classification performance of a classifier with respect to some test data." (Ting, 2011). Computing the confusion matrix of the face detections can provide a clear view for the performance of facial recognition and analysis software. To find the true positive face, the concept of Jaccard Index is used. Jaccard Index is calculated as the overlapping between two areas (Rosebrock, 2016). The script of ConfusionMatrix.py is implemented to compute the convex hulls of the human annotated points and the machine annotation points, and the Jaccard Index between the two hulls (i.e., human annotated vs. machine detected). The computation of convex hull of various objects overlap is widely used in computational geometry field (Avis, et al, 1997), it is used to compute the Jaccard Index in this work. A face is regarded as a true positive face detection when the Jaccard Index value is greater than or equal to 0.5.
Taking the Figure 2 below as a sample, the blue x marks are the annotated landmark coordinates, collectively they form a convex hull (shown in green). Similarly, the pink area covering the red dots is the convex hull of detected landmark coordinates after conversion. Therefore, based on the definition above, the insertion of pink area and green area / their union is large than 0.5, so this face would be marked a true positive face match.

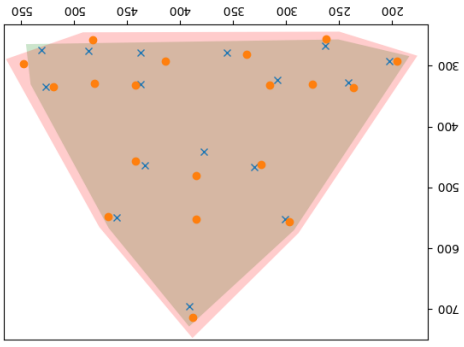

Figure 2. The Insertion of Convex Hull Area from Human Annotated and Machine Detected Landmark Coordinates.

This method may adversely affect the performance score for OpenFace, because its face detection always fills in the missing face landmarks when the face is rotated (see Figure 3). A human viewer would consider a face has been found in this example, however, due to the landmarks annotated in the output do not match well with the answer, we consider this a false positive.

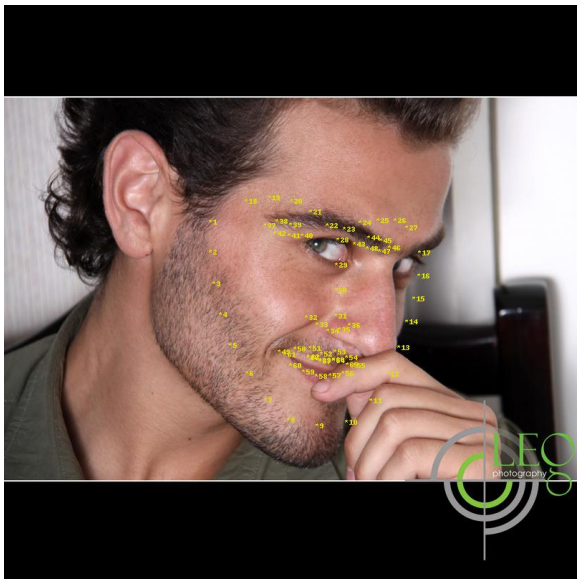

Figure 3. OpenFace Auto Fills Missing Points in Landmark Detection.

Based on that decision, there are three values recorded (TP, FP, FN) in the confusion matrix. And they help us calculate F1 score of confusion matrix. The formula of F1 is F1 =2*TP/(2*TP+FP+FN). TP means the number of the correctly detected faces in this dataset, the FP is the number of false detections in this dataset, FN is used to represent the missed detections of faces in dataset. Based on the rule above, detection scores (1 or 0) were generated for each face in images. Figure 4 shows the detection information produced by the program, which includes an image annotation file (one annotation file describes one face), the TP value (1 means detected, 0 means undetected), and its corresponding detection file.

300w_deva_ConfusionTable.csv

~/allData_deva/300w_0.5/300w_deva_ConfusionTable.csv

```
ImageID,TP,landmarks_file_ID_pridiction
indoor_001_face_anno_01,1,indoor_001_face_det_0
indoor_002_face_anno_01,1,indoor_002_face_det_0
indoor_003_face_anno_01,1,indoor_003_face_det_0
indoor_004_face_anno_01,1,indoor_004_face_det_0
indoor_005_face_anno_01,1,indoor_005_face_det_0
indoor_006_face_anno_01,1,indoor_006_face_det_0
indoor_007_face_anno_01,1,indoor_007_face_det_0
indoor_008_face_anno_01,1,indoor_008_face_det_0
indoor_009_face_anno_01,1,indoor_009_face_det_0
indoor_010_face_anno_01,0,/
indoor_011_face_anno_01,1,indoor_011_face_det_0
indoor_012_face_anno_01,1,indoor_012_face_det_0
indoor_013_face_anno_01,1,indoor_013_face_det_0
indoor_014_face_anno_01,0,/
indoor_015_face_anno_01,1,indoor_015_face_det_0
indoor_016_face_anno_01,1,indoor_016_face_det_0
indoor_017_face_anno_01,0,/
indoor_018_face_anno_01,1,indoor_018_face_det_0
indoor_019_face_anno_01,1,indoor_019_face_det_0
indoor_020_face_anno_01,0,/
indoor_021_face_anno_01,1,indoor_021_face_det_0
indoor_022_face_anno_01,1,indoor_022_face_det_0
indoor_023_face_anno_01,1,indoor_023_face_det_0
indoor_024_face_anno_01,1,indoor_024_face_det_0
indoor_025_face_anno_01,0,/
```

Figure 4. The Sample of Confusion Matrix Table Generated.

3.2 **Mean Error of Euclidean Distance**

The confusion matrix can tell us whether the program detected the face in the right place, but we also need to use the method of Mean Error of Euclidean Distance (Vlachos, 2011) to check the accuracy of detected landmarks in the faces for all true positive detections.

Interocular distance is the distance between the outer corners of the left and right eyes, for example, the point 37 and the point 46 in MultiPIE (as Figure 5 showed), see formula below.

$$Interocular\ Distance = \sqrt{(y_{46} - y_{37})^2 + (x_{46} - x_{37})^2}$$

Interocular Distance is then used in the computation of the Mean Error of Euclidean Distance between human annotation and machine detection in one face. The Mean Error of Euclidean Distance is defined as the sum of point to point Euclidean distances between annotated landmark coordinates and the detection landmark coordinates of a true positive face normalized by the product of the number of points and the interocular distance (the interocular distance is used to normalize all error metrics). The formula is represented as following, N represents the number of landmark coordinates. i_a represents the annotation coordinates and i_d means the detection coordinates,

$$\text{Mean Error of Euclidean Distance of one face} = \frac{\sum_i^N \sqrt{(y_{i_d} - y_{i_a})^2 + (x_{i_d} - x_{i_a})^2}}{N * Interocular\ Distance}$$

And it is necessary to apply this formula in whole dataset (600 images) to get the accuracy performance for the true positive face. The formula is displayed below. In here, the M represent the number of images.

$$\text{Mean Error of Euclidean Distance over a collection of images} = \frac{\sum_j^M \frac{\sum_i^N \sqrt{(y_{i_d j} - y_{i_a j})^2 + (x_{i_d j} - x_{i_a j})^2}}{N * interocular\ distance}}{M}$$

The script FaceAccurateCalculator.py is designed to do this once for all of the annotated points in the "face" calculation. In addition, it can do the same calculation for just the outer corners of the eyes, which I call the "eye" calculation. In this case, the all points are replaced with only the two points of outer corners of the eyes (the point 37 and the point 46 as mentioned above).

Figure 5. The 68-point Mark-up Scheme Developed at CMU (MultiPIE), the Figure Taken from I-bug Website (Sagonas et al, 2016).

4 Results

Table 1. Confusion Matrices of OpenFace, Deva and Face++ Face Detection

		OpenFace			Deva			Face++	
		Positive	Negative		Positive	Negative		Positive	Negative
300w	TRUE	549	/	TRUE	532	/	TRUE	579	/
	FALSE	0	51	FALSE	35	68	FALSE	2	21
	F1	0.955613577		F1	0.911739503		F1	0.980524979	
		Positive	Negative		Positive	Negative		Positive	Negative
AFLW	TRUE	374	/	TRUE	388	/	TRUE	631	/
	FALSE	194	326	FALSE	134	312	FALSE	149	69
	F1	0.589905363		F1	0.63502455		F1	0.852702703	

Table 1 shows that Face++ has the best F1 score in both datasets, this means it has the small amount of false detection and missed detection. Besides, the OpenFace is better than Deva in 300w dataset, while in AFLW, the Deva is better than OpenFace.

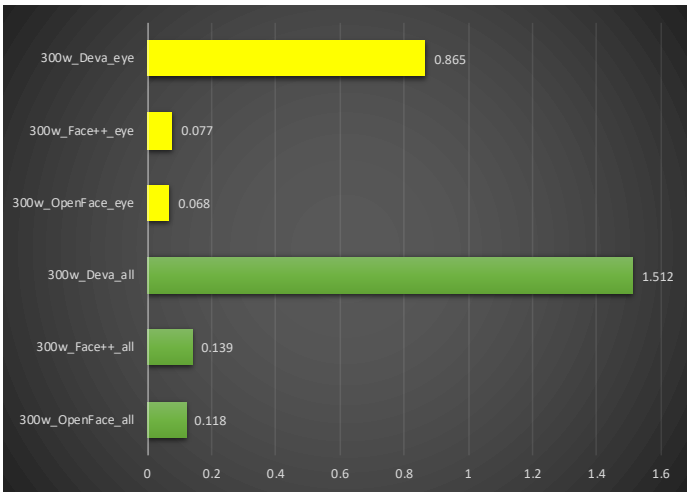

Figure 6. Median Normalized Errors of True Positive Faces for 300w Dataset (Lower Error is Better).

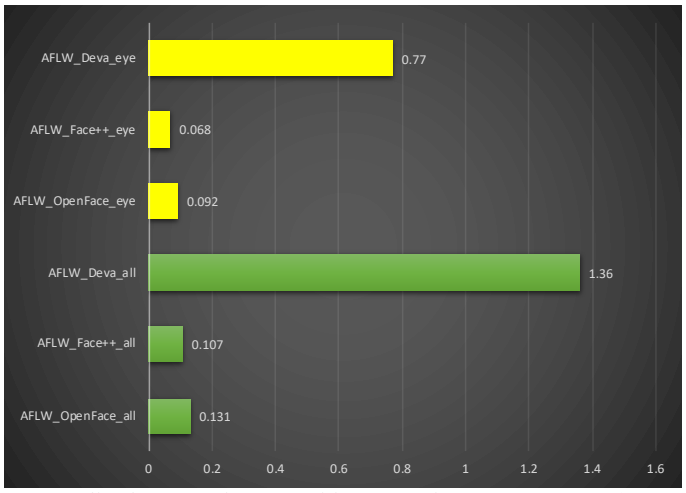

Figure 7. Median Normalized Errors of True Positive Faces for AFLW Dataset (Lower Error is Better).

The Figure 6 and Figure 7 above indicate the accuracy of these faces in particular points. According to these results, it is clearly that OpenFace has a smaller median normalized error than Face++ in 300w, while Face++ performs better in AFLW. Deva has much larger error than Face++ and OpenFace.

5 Discussion

Based on the above result, it is noticed that in terms of the F1 performance, the three programs we tested have a big difference in AFLW dataset. Thus, the cause of performance differences in AFLW is further explored below. It is hypothesized that the face rotation can lead to the difficulty in face detection. Therefore, the images of AFLW dataset where the faces are undetected by facial recognition software are examined using two way, the first one is to examine the performance on the faces rotated by mathematical x axis or z axis.

5.1 Face Rotated by Mathematical X Axis

The angle of face rotated by mathematical x axis is represented using the green lines in Figure 8. To be motioned, the angle of face is considered as 0 degree when two eyes are horizontal.

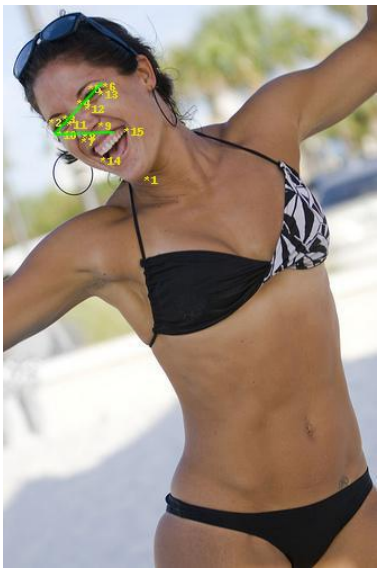

Figure 8. Sample Image about the Angle of Face Rotated by Mathematical X Axis

The script slope.py is implemented to compute and examine the angles of the undetected faces.

Table 2. The Distribution of Undetected Faces Rotated by Mathematical X Axis in OpenFace Detection

OpenFace		
Rotation Degree Range	Number(undetected/total)	Percentage
0-14	121/432	28%
15-19	48/91	52%
30-44	15/31	48%
45-59	3/3	100%
60-74	2/2	100%
75-90	137/141	97%

The Table 2 above indicates the number of undetected images which is rotated by mathematical x axis for certain degree. In addition, this table shows the distribution of face rotation by mathematical x axis in whole dataset. The rotation degrees in the range of 45-59 and 60-74 have only 3 faces in whole dataset, such that this rotation angle will not be discussed in all three programs. In OpenFace, it is clearly found that the rotation angle in the range of 75-90 has significant influence on the OpenFace Detection. There are 137 faces are undetected in total 141 faces of this dataset in the rotation degree range.

Table 3. The Distribution of Undetected Faces Rotated by Mathematical X Axis in Deva Detection

Deva		
Rotation Degree Range	Number(undetected/total)	Percentage
0-14	123/432	28.50%
15-19	50/91	55%
30-44	29/31	93%
45-59	2/3	66%
60-74	2/2	100%
75-90	106/141	75%

Similarly, In Deva (see Table 3 above), when the rotation angle in the range of 30-44 and 75-90, the undetected face percentage reach at 93% and 75 % respectively. These rotation angles bring the difficulty to Deva detection. In Face++ (see Table 4 below), there is no specific rotation angle range which result in the difficulty for detection.

Table 4. The Distribution of Undetected Faces Rotated by Mathematical X Axis in Face++ Detection

Face++		
Rotation Degree Range	Number(undetected/total)	Percentage
0-14	24/453	5.60%
15-19	11/91	12.10%
30-44	3/31	9.70%
45-59	0/3	0%
60-74	1/2	50%
75-90	30/141	21.30%

In another word, the angle of face rotated by mathematical x axis has affect the facial detection, the larger angle of face rotation leads to the higher the number of undetected faces in Deva and OpenFace. The Face++ perform better than Deva and OpenFace.

5.2 Face Rotated by Mathematical Z Axis

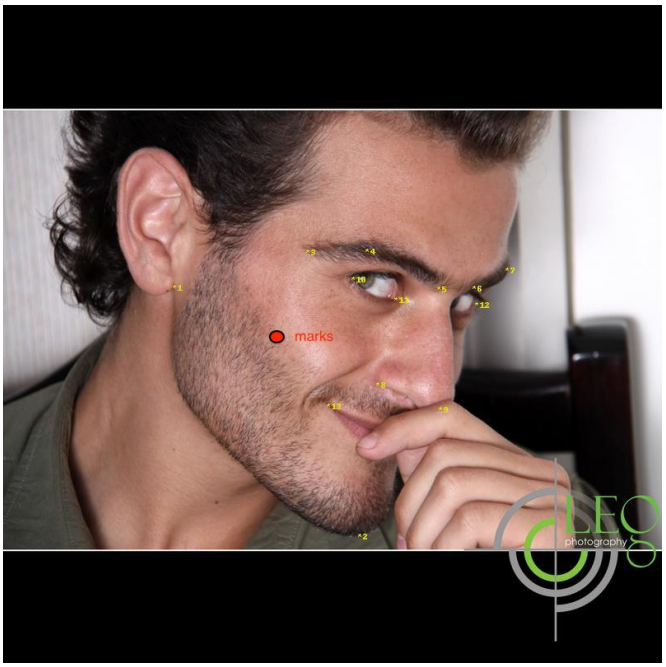

Figure 9. Sample Face Which is Rotated by Mathematical Z Axis about 80 Degrees with 13 Landmark Coordinates.

To find the rotated face by mathematical z axis, the undetected faces' landmark coordinates in annotation files whose number is less than 11 are computed. The faces which have less than 11 annotation landmark coordinates mean they are half faces due to the fact that the AFLW landmark scheme initializes 21 landmark coordinates for whole face. The less landmark coordinates mean that the larger angle of face rotated by mathematical z axis, because the face rotation by z axis can lead to the missing of landmark coordinates. With the Figure 9 as an example, this face is rotated by mathematical z axis about 80 degree, and the face still keeps 13 landmark coordinates. However, if the face is rotated by mathematical z axis 90 degree or more, this face will lose some landmark coordinates, and then the number of the landmark coordinates will be less than 11.

After the computation using the script invalid_statistics.py, the Table 5 below is generated as following.

Table 5. The Number of Half Faces in Undetected Faces and the Percentage

	OpenFace	Deva	Face++
number of half faces in undetected faces	249/326	191/312	46/69
percentage	76.4%	61%	67%

A conclusion is drawn that the oblique and 90-degree profile views of faces (half faces) lead to the difficulty and error increased in landmark location detection, because these faces account for more than 60% of undetected faces, and this rate can reach more than 70% in OpenFace.

5.3 The Monochromatic Images

Using the script isGreyScale.py to determine whether the images are color images or monochromatic images. The Table 6 is generated.

Table 6. The Distribution of Correctly Detected Monochromatic Images and Color Images.

	OpenFace	Deva	Face++
color detection rate	53.2% (269/506)	53.4% (270/506)	90% (456/506)
mono detection rate	34% (32/94)	46% (44/94)	90%(85/94)

Based on the Table 6 above, it is clearly found that the monochromatic (black/white) images is a challenge for OpenFace and Deva. The color images detection rate in both OpenFace and Deva have around 53%. It means that there are about 270 color images are correctly detected in total 506 color images in dataset, while the monochromatic images detection rates in OpenFace and Deva are 34% and 46% respectively. Therefore, OpenFace has low detection rate for monochromatic images.

6 Conclusion

Through careful data transformation and error analyses, the following conclusions may be drawn from this performance comparison study:

All three facial programs work fine for the detection of images where the face is looking nearly directly at the camera. The oblique and 90-degree profile views of faces and monochromatic conditions have different impacts on the three tools.

Firstly, the incline angle of face rotated by mathematical x axis do harm to facial detection, especially for large rotation angle. Deva has low detection rate in angle range 30-44 and 75-90, the OpenFace has low detection rate in angle range 75-90. Face++ is less affected relatively speaking.

Secondly, the faces rotated by mathematical z axis more than 90 degrees (“half face”) cause low detection rate on facial detection for all three systems tested.

Thirdly, the images with some features such as monochromatic images pose threat to face detection on OpenFace and Deva.

Therefore, the Face++ has the best performance in different imaging condition.

In the future work, it can be a good proposal to explore how closed eye affect face detection. The dataset I considered doesn’t contain faces with closed eyes.

References:

T. Baltrusaitis, P. Robinson, and L.-P. Morency, “Openface: an open source facial behavior analysis toolkit,” in Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, pp. 1–10, IEEE, 2016.

T. Baltrusaitis, P. Robinson, and L.-P. Morency, “Constrained local neural fields for robust facial landmark detection in the wild,” in Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, pp. 354–361, IEEE, 2013.
A. Zadeh, T. Baltrusaitis, and L.-P. Morency, “Convolutional experts constrained local model for facial landmark detection,” in *Computer Vision and Pattern Recognition Workshops*2017.
OpenFace: https://github.com/TadasBaltrusaitis/OpenFace
Beijing Kuangshi Technology Co. (2019). What is Face Cognitive Services? Retrieved April 25, 2019, from https://www.faceplusplus.com/about-us.
X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2879–2886, IEEE, 2012.
Deva: http://www.cs.cmu.edu/~deva/papers/face/index.html
C. Sagonas, E. Antonakos, G, Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 faces In-the-wild challenge: Database and results. Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild". 2016.
C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. A semi-automatic methodology for facial landmark annotation. Proceedings of IEEE Int’l Conf. Computer Vision and Pattern Recognition (CVPR-W), 5th Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2013). Oregon, USA, June 2013.
C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge. Proceedings of IEEE Int’l Conf. on Computer Vision (ICCV-W), 300 Faces in-the-Wild Challenge (300-W). Sydney, Australia, December 2013.
Martin Koestinger, Paul Wohlhart, Peter M. Roth and Horst Bischof. Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies, 2011.
Ting K.M. (2011) Confusion Matrix. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA
Rosebrock, A. (2016, November 7). Intersection over Union (IoU) for object detection. Retrieved April 24, 2019, from https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/
Vlachos M. (2011) Similarity Measures. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA
Avis, et al. “How Good Are Convex Hull Algorithms?” Computational Geometry: Theory and Applications, vol. 7, no. 5, 1997, pp. 265–301.

Competency

C1: Computational and analytic thinking and doing: Students will establish the ability to exercise the four key techniques of computational thinking: decomposition, pattern recognition, abstraction, and algorithms in solving information and data challenges, in addition to analytically.

C1.A: Decomposition: Students will be able to break down a complex problem or system into smaller, more solvable problems.

To evaluate the performance of facial recognition and detections software, I break down the performance problem into two smaller problems (1. the problem of face recognition rate 2. the problem of face accuracy rate). To evaluate the influence of face rotation in missed detection, I calculate and count the number of faces in missed detection about their face rotation by mathematical x axis and z axis.

C1.B: Pattern recognition: Students will be trained to look for similarities among and within problems.

I learned the concept and knowledge of confusion matrix and Euclidean distance, therefore, I use the confusion matrix to visualize of the performance of an model, I also use mean error of Euclidean distance to compute the point-to- point difference among the face from annotation and the face from program detection.

C1.C: Abstraction: Students will gain the ability of recognizing and focusing on the essential components of a problem/issue while ignoring distracting peripheral factors in order to develop one solution that works for a class of problems.

I know there are many distracting peripheral factors in face detection, but I think the face rotation is the most important cause, such that I focus on analyzing the influence of face rotation firstly.

C1.D: Algorithms: Students will be able to design and implement a step-by-step solution to a problem, including design and implement a computer algorithm using a computer language to solve a problem.

I design and implement the algorithms of confusion matrix, which contain the method Jaccard Index (insertion of convex hull area of two face /union of convex hull area of two face) to determine whether the face is correctly detected, and I use the algorithm of mean error of Euclidean distance to find the normalized error between the face from annotation and the face from detection.

C1.E: Students will demonstrate fluency in at least one programming language.

In this project, I almost use python as main programming language, besides, I uses bash script also.

C2: Data manipulation, analysis, and interpretation: Students will obtain the skills of collecting, manipulating, and analyzing different types of data at different scales, and interpreting the results properly.

C2:A: Students will be able to identify specific types of data for different analytical methods

I use Jaccard Index (the insertion of convex hull area of two faces / the union of convex hull area of two faces) to compute convex hull area which is consisted of the face landmark coordinates. Also, I the face landmark coordinates to compute the mean error of Euclidean distance.

C2:B: Students will be able to use/develop efficient computational methods to clean, format, transfer, and store data.

1.I extract the annotation files from AFLW database. 2. I convert the annotation files and detection files which store the landmark coordinates into the dataset landmark scheme.

C2:C: Students will be able to apply appropriate statistical, machine learning, visual analytics, and other techniques to identify patterns and make sound predictions with given data.

1.I statistical the distribution of landmark points for the undetected face to analyze the influence of face rotation. 2. I visual the landmark coordinates on images to find the feature of images.

C2:D: Students will be able to develop methods to align and integrate data from multiple sources.

1. I convert the detection files which store different landmark coordinates into the dataset landmark scheme.

C2:E: Students will understand the ethical and legal requirements of data privacy and security.

The datasets and software are legal and get the permission from the owners.

C3: Communication and teamwork: Students will acquire skills to work with others within and across disciplines and be effective communicators.

C3.A: Students will acquire experience working in an interdisciplinary team, either as a productive team member or a team leader. Students will become effective project managers.

This is a team project of IVILab leading by UA computer vision professor Kobus, I am a team leader of landmark group. In early work, our landmark group member help collect the data, and get the API works etc.

C3:B: Students will be able to effectively articulate various evidence supporting a solution and to communicate the results of their work, using appropriate graphics, visualizations, multi-media vehicles, or artistic performance.

I visualize the landmark points on images, the visualization work can help us the recognize the feature of images

C4: Creative contributions: Through experiential learning, students will know how to conduct original and innovative work, involving computational thinking, data-intensive methodologies, and/or human-centered designs that will extend the body of knowledge in the field of Information.

The Face landmark detection is popular now, the performance of Face detection attracts audience's interest. I am the first one to compare the performance of Deva, OpenFace, Face++ (three different type of face detection model) in face recognition and detection face accuracy. Besides, I explore the performance differences and analyze the influence of face rotation to face detection.

C5: Ethics and Values: Students will demonstrate an understanding of information/data ethics, and the values of the information fields to serve diverse user groups.

The dataset and software are legal and get the permission from the dataset and software owner.