# Face Pose Detection

June 2021

Nawaf Alageel

# Abstract

This paper addresses a simple face pose detection from the perspective face recognition algorithm (open set recognition) that requires frontal faces. The algorithm detects the faces if it is a frontal face or rotated to the right or left (i.e., right profile, or left profile). Most of the face analysis algorithms (e.g., gender, and expression) require frontal up-right faces because it has an important role in classification methods. Our algorithm detects only out-of-plane orientation estimation which is face rotation with respect to the y-axis. The detection algorithm is the core of this technique, MTCNN is the model that is used to detect faces. When a face is detected for the first time, we extract face landmarks (i.e., right eye, left eye, nose, left mouth, and right mouth) and from these points, we calculate the angels between the right eye, left eye, and the nose using a Cartesian coordinate system for euclidean space 2D. By setting threshold ranges for the right eye angle and left eye angle we can estimate if the face is rotating to the left, right, or frontal face. We evaluate our experiments on one dataset which is Pointing Head Pose Image Database (HPID), and due to lack of benchmarks dataset with proper annotation we only benchmark on one dataset. Moreover, the accuracy of 91.19% was achieved on the HPID dataset.

# List of Figures



# List of Tables



# Contents

# 1 Introduction

In applications such as face recognition systems depend not only on the identity of the person but also on parameters such as head pose, illumination, facial expression. Variations in pose might be the critical one among all the parameters especially if the face recognition model depends on an open set approach for recognition. Another approach that uses a classification model for recognition and the system differs for open set approach with the last stage which is the classifier at the end, DeepFace [1] is one of the popular systems following the face classifier approach. However, face classifiers also need to be invariant to pose, illumination, expression, and image quality [1]. Historically, there have been several approaches indistinguishable from the DL approach, Jones and Viola [2] also trained a decision tree to classify an image window as left or right profile, which might be the closest methodology to us. Our proposed pose estimation only includes three poses frontal face, left, or right profile. The challenges presented by [3] are three: First, the commonly used landmark-based face model assumes that all the landmarks are visible and is therefore not suitable for profile views. Second, the face appearance varies more dramatically across large poses, ranging from frontal view to profile view. Third, labeling landmarks in large poses is extremely challenging since the invisible landmarks have to be guessed. This work aims to address the first gap in the literature of the first challenge. We argue that complicated methods it might rely entirely on data, and neural network. What we present an elegant and robust way based on the detection model. However, many related works follow the first approach. In section 2 is where we discuss related work.

## 2 Related Work

There are several algorithms for pose estimation. And most of the algorithms are beyond the scope of this paper since most of them consider 3D pose estimation. We will explain the details of some of the algorithms. Starting with DL (Deep Learning) approach, Hopnet [4] they present an elegant and robust way to determine pose by training a multi-loss convolutional neural network on 300W-LP dataset [3], a large synthetically expanded dataset, to predict intrinsic Euler angles (yaw, pitch and roll) directly from image intensities through joint binned pose classification and regression. Similarly, Chang et al. [5] proposed a simple CNN, uniquely trained to regress 6DoF face pose, directly from image intensities. All these approaches differ from our work since we directly calculate the angles between three points from the facial landmark the model produced and the idea of predicting only three labels. Huang et al. introduced an approach for the problem of face pose discrimination using Support Vector Machines (SVM), SVM model classifies three possible face poses (i.e., Frontal, right, and left) the part where we differed in their labels they define a degree on both sides and that degree equals 33.75°. The SVM achieved perfect accuracy 100% discriminating between the three possible face poses on unseen test data. In our case, their work is indistinguishable from our work except for the model part where we do not use a model to predict the face pose, we either use threshold values from the calculated angles from face landmarks that the detection model produces. In contrast, Jones and Viola [2] trained a decision tree to classify an image window as a left or right profile. It achieves 95.4% accuracy on the training set. Since they have only two classes, in this case, the decision tree is only useful to test its effect on the detection rate and false positive rate and not to improve the detector's speed. Thus, they confirm that the ROC curves for the two-stage detector and the try-both-profiles detector were close as in the non-upright face case. These can be very accurate. In section 3 is details of the approach.

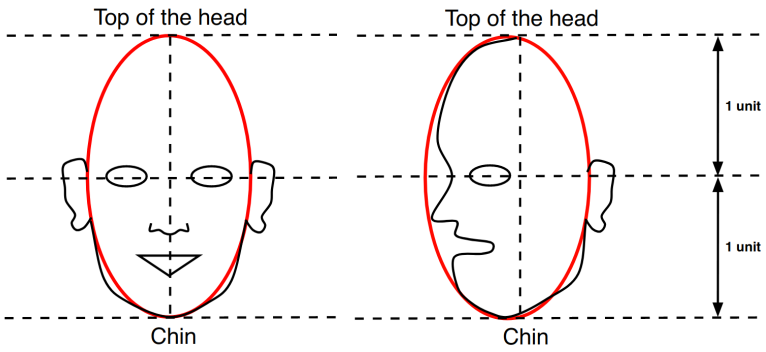
# 3 Proposed Method

We mentioned in the previous sections that we are using MTCNN as the detection algorithm which is the core of our work, but actually there are various detection algorithms that could be replaced with what we selected. We used a dataset that could be found publicly [6] to illustrate and evaluate our methodology. The dataset consist of 6660 images of 90 subject and each subject has 74 images with poses took every 5 degrees from frontal face 0° to right and left profile ±90°. But they did not mention if the face is a right profile or left profile. On the other hand, our requirement for annotation should include this information (i.e., frontal face, right or left profile ) which is only three classes is missing, so Jain and Learned-Miller proposed guidelines for annotating faces using ellipses [7] to determine if the face is frontal, side profile, and tilted faces. We labeled that data based on Jain and Learned-Miller guidelines and we set new guideline to determine the right and left profiles based on the degree of the face. And here is the guideline we use:

- Right profile defined from $+20°$ to $+90°$.
- Left profile defined from $-20°$ to $-90°$.
- For frontal face labeling we follow [7] guidelines:

  1. If major axis parallel to the nose.
  2. When the chin to the bottom end of the major axis of the ellipse.
  3. Make the eyes align with the minor axis of the ellipse.
  4. Ensure that the ellipse traces the boundary between the ears and the face.

To illustrate more see figure 1 [7].




(a) Frontal Face. (b) Side Face.

Figure 1: Face Guides.

## 3.1 Problem formulation

Firstly, our proposed method used a landmark-based face model that assumes that all the landmarks are visible. After the model produces the landmarks we draw a line between three points i.e., right eye, left eye, and the nose. Forming a triangle and through that triangle, we calculate the angle $\theta_1$ and $\theta_2$. This step, geometric object that possesses both a magnitude and a direction. A vector can be pictured as an arrow or a line. Its magnitude is its length, and its direction is the direction that the line points to. The magnitude of a vector a is denoted by $||a||$. The dot product of two Euclidean vectors a and b is defined by, where $\theta$ is the angle between a and b. Illustration can be seen in figure 2. Additionally, the dot product may be defined geometrically. The geometric definition is based on the notions of angle and distance (magnitude of vectors). The equivalence of these two definitions relies on having a Cartesian coordinate system for Euclidean space 2D. An example of how to calculate angles from three points are provided in figure 3 and the equations from (1)-(4).

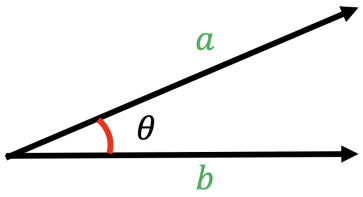



Figure 2: Angles Representation.

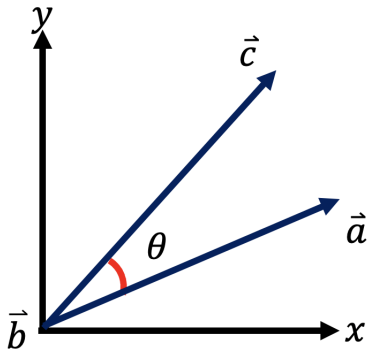



Figure 3: Angle Example.

$$\overrightarrow{\text{ba}} = a - b \tag{1}$$

$$\overrightarrow{\text{bc}} = c - b \tag{2}$$

$$\cos(\theta) = \frac{(\overrightarrow{\text{ba}} \cdot \overrightarrow{\text{bc}})}{(|\overrightarrow{\text{ba}}| \cdot |\overrightarrow{\text{bc}}|)} \tag{3}$$

$$\theta = \arccos(\frac{(\overrightarrow{\text{ba}} \cdot \overrightarrow{\text{bc}})}{(|\overrightarrow{\text{ba}}| \cdot |\overrightarrow{\text{bc}}|)}) \tag{4}$$

## 3.2 Method

In this section, we explain how we can use the angles to find the face pose. Face orientation, or pose, is determined by 2 angles, $\theta_1$ and $\theta_2$. The details of angles on top of a face is shown in figure 4. After the model detected the face and the landmarks, we further look into the produced landmarks, and calculating the angle between the vector $||rl||$ and $||rn||$, similarly between the vector $||rl||$ and $||ln||$. Hence, we produced both $\theta_1$ and $\theta_2$. After calculating the angles, $\theta_1$ and $\theta_2$ act as an input to the algorithm, if the value of $\theta_1$ and $\theta_2$ between certain thresholds then we consider the detected face as frontal pose. Otherwise if $\theta_1 < \theta_2$ then it is left profile and the opposite is the right profile. The threshold ranges is calculated from the angles we produced, and then the values consider as thresholds to the algorithm shown in Algorithm 1, and we denotes $\theta_1$ ranges as $\alpha_1 = \mathbb{S}_1$ and $\{\alpha_1 \subset \mathbb{R}\}$ which is a range of values and $\theta_2$ as $\alpha_2 = \mathbb{S}_2$ also $\{\alpha_2 \subset \mathbb{R}\}$. In the section 4.2 material, we presented a method on how to select and set the thresholds and the limitations.

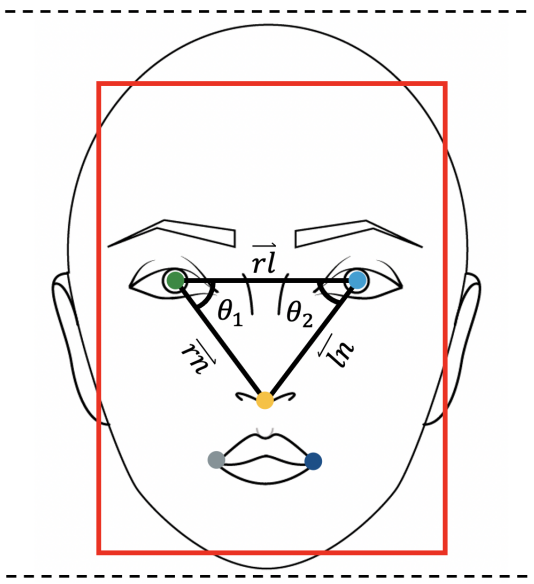



Figure 4: Face Angles.

**Algorithm 1:** Detect Face Orientation or Pose

**Result:** face orientation or pose

**if** *Face Detected* **then**
  Calculate angles from face landmarks;
  **if** $\{\theta_1 \exists \alpha_1\}$ *and* $\{\theta_2 \exists \alpha_2\}$ **then**
    Detected as Frontal Face;
  **else if** $\theta_1 > \theta_2$ **then**
    Detected as Right Profile;
  **else**
    Detected as Left Profile;
  **end**

## 3.3 Proposed Model for Detection

Multi-Task Cascaded Convolutional Neural Network (MTCNN) [8] is the model we use for detection. The model did very well with detecting non-frontal faces, and it detects small faces, also it proposed landmarks detection, and it considers fast detector compared to other models or algorithms, see 1 for speed results. The model consists of three stages:

1. It proposes candidates windows quickly through shallow CNN, this stage called P-Net.

2. All candidates from P-Net are feed to another complex R-Net which further rejects a large number of non-faces windows.

3. Lastly, it uses another CNN similar to P-Net to verify the results window and output five facial landmarks positions through O-Net.

| | Implementation | Backbone | Avg Time (sec/img) | Accuracy |
|---|---|---|---|---|
| Haar Cascade | - | - | 0.055 | 62.5% |
| MTCNN | Keras | - | 2.4 | 96.42% |
| | Pytorch | - | 0.085 | 96.42% |
| HOG | Dlib | - | 0.3 | 60.71% |
| | Dlib & other | - | 0.2 | 71.42% |
| CNN | Dlib | - | 0.3 | 60.71% |
| | Dlib & other | - | 0.2 | 89.28% |
| RetinaFace | MXNet | ResNet | 1.24 | 92.85% |
| | MXNet | MobileNet | 0.515 | 78.57% |
| | Pytorch | ResNet | 1.0975 | 100% |
| | Pytorch | MobileNet | 0.205 | 96.42% |

Table 1: Comparison between Detection models and algorithms.

The reason why the MTCNN model did well with small faces because of the approach Zhang et al. [8] uses, they resize the input image to different scales to build an image pyramid, and this images pyramid feed into three networks. There are many other detection models for instance RetinaFace [9], and even simpler algorithms such as Haar cascade [10], and HOG Based for face detection. See table 1 for performance of many detection algorithms. We used a local dataset to evaluate each model, the data consist of 120 images.

# 4 Experiments

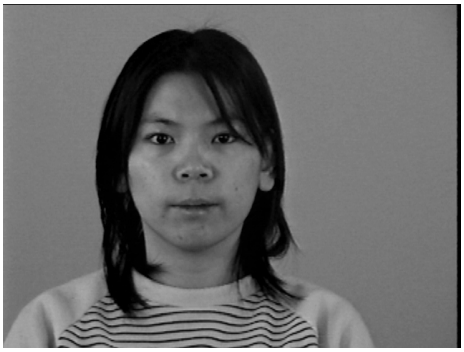
## 4.1 Dataset and Data Augmentation

Unfortunately, for face pose datasets there are few. To our knowledge, the process of annotating such problems as face orientation is considered a subjective matter. However, we found several sources for datasets with face pose notations either for tilt, or pan. But never a dataset with a notation to indicate if the face is frontal, or rotated to the right or left. The dataset is used to demonstrate the algorithm, and to select the appropriate threshold for the angles as mentioned in 3.2. And the problems with detection not to evaluate it. As we mentioned before the dataset consist of 6660 images of 90 subject and each subject has 74 images with poses took every 5 degrees from frontal face 0° to right and left profile ±90°. But they did not mention if the face is a right profile or left profile. In each degree, there are 360 images. After we implemented our labeling technique we result in a total of three classes: 1- Frontal 2- Right 3- Left. For Frontal, there are 1440 images, 2520 for both right and left. For more information about the data set see table 2. These results show us many things to consider, drawbacks of the algorithm, or limitations.

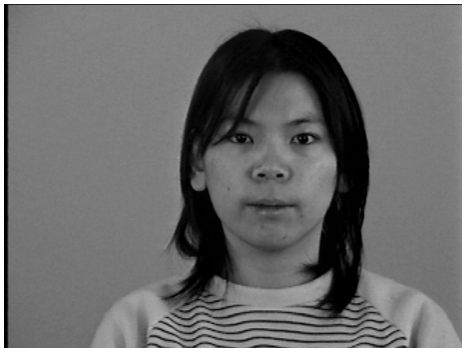
| Degree | # of images | # of detection | Detection accuracy |
|---|---|---|---|
| 0° | 180 | 180 | 100.00% |
| ±05° | 360 | 360 | 100.00% |
| ±10° | 360 | 360 | 100.00% |
| ±15° | 360 | 360 | 100.00% |
| ±20° | 360 | 360 | 100.00% |
| ±25° | 360 | 359 | 99.72% |
| ±30° | 360 | 359 | 99.72% |
| ±35° | 360 | 359 | 99.72% |
| ±40° | 360 | 358 | 99.44% |
| ±45° | 360 | 355 | 98.61% |
| ±50° | 360 | 343 | 95.27% |
| ±55° | 360 | 341 | 94.72% |
| ±60° | 360 | 330 | 91.66% |
| ±65° | 360 | 315 | 87.50% |
| ±70° | 360 | 290 | 80.55% |
| ±75° | 360 | 255 | 70.83% |
| ±80° | 360 | 211 | 58.61% |
| ±85° | 360 | 177 | 49.16% |
| ±90° | 360 | 151 | 41.94% |
| All | 6660 | 5823 | 87.74% |

Table 2: MTCNN result with ourDatabase

Data augmentation is a technique that we create new data based on modifications of our existing data so essentially we are creating new augmented data by making reasonable modifications to the training data that we have. Data augmentation is already provided from the dataset owners, they claimed there are two types of images, first original image, second synthesized image. It is only flipping horizontally see figure 5.



(a) Subject 44 Original Image.



(b) Subject 44 Synthesised Image.

(c) Subject 39 Original Image.

(d) Subject 39 Synthesised Image.

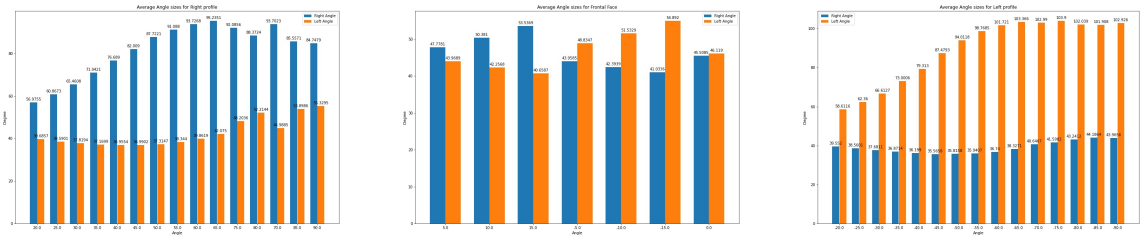
Figure 5: Data Augmentations

## 4.2 Threshold Selection

Due to difficulties in estimating the threshold angles for $\alpha_1$ and $\alpha_2$, we have extract minimum and maximum angle values we calculated from ourDatabase in the defined ranges as a frontal face. As we need two positions for eyes to estimate the face pose, the face pose can not be prediction accurately or assign an accuracy. Furthermore, the fact that people do not have the same distance between eyes makes the prediction of the position of the other eye inaccurate for the model. The threshold as we mentioned earlier is based on the minimum and maximum angle values we calculated from ourDatabase and these values are produced by the MTCNN model, and as a result, the numbers we get from the model not only limits our method but also might cause a crucial problem in predicting the face pose. Nevertheless, introducing a new detection model might be required to adjust the threshold selection. For us, we set the thresholds to $\alpha_1 = [35, ..., 57]$ and for $\alpha_2 = [35, ..., 58]$, thus the prediction would be

$$Pose\_Prediction = \begin{cases} FrontalFace, & if\ 35 \leq \theta_1 \leq 57 \text{ and } 35 \leq \theta_2 \leq 58 \\ RightProfile, & if\ \theta_1 > \theta_2 \\ LeftProfile, & Otherwise \end{cases} \tag{5}$$

We also provide the angle values of each label we define (i.e., frontal face, right profile, left and profile) in Figure 6.



(a) Right Profile Angles (b) Frontal Face Angles (c) Left Profile Angles

Figure 6: Angle Values From ourDatabase

## 4.3 Results on Threshold Selection

Our informal experiment on ourDatabase lend to support our threshold selection, as shown in figure 7 is the confusion matrix. In our experiment, we achieved 92% accuracy, 94% precision, 93% recall, and 93% f1-score. Figure 8 shows some of the results from the experiment. Although this seems good results, this process makes the results biased and inaccurate, since we selected the threshold from ourDatabase. However, next section 5 we discuss the results on a different dataset to evaluate our methodology accurately.

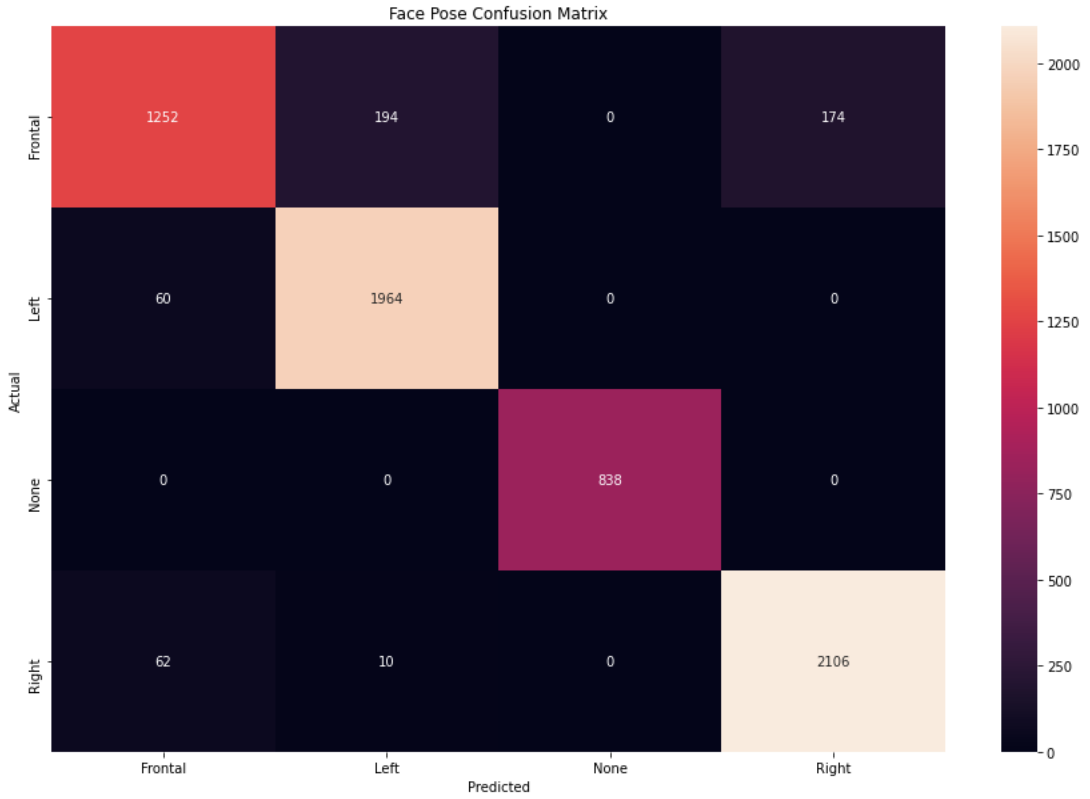
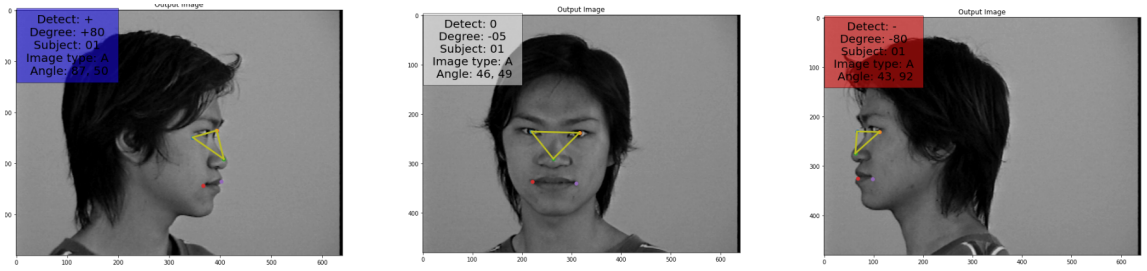
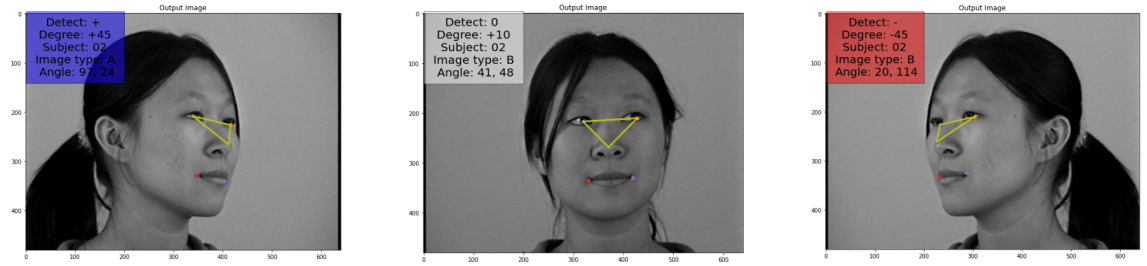


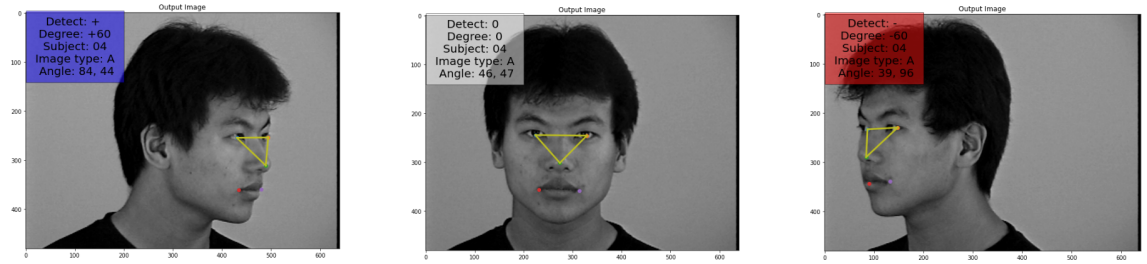

Figure 7: ourDatabase Confusion Matrix.




(a) Subject01 Right Profile (b) Subject01 Frontal Face (c) Subject01 Left Profile




(d) Subject02 Right Profile (e) Subject02 Frontal Face (f) Subject02 Left Profile

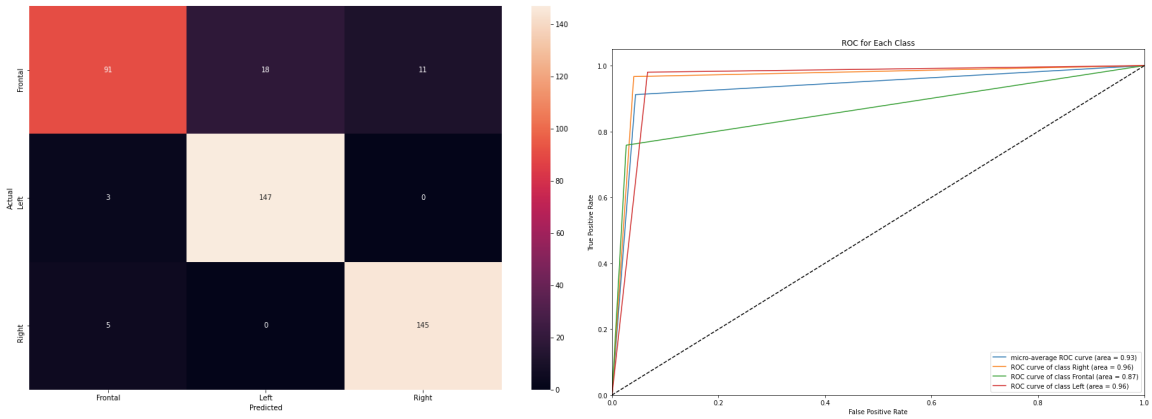



(g) Subject04 Right Profile (h) Subject04 Frontal Face (i) Subject04 Left Profile

Figure 8: ourDatabase Results

# 5 Results

This Section evaluates our algorithm on Pointing Head Pose Image Database [11], the dataset consist of 2790 monocular face images of 15 persons with variations of pan and tilt angles from -90 to +90 degrees, but since tilt is out of the scope of this work we eliminate any tilt in the dataset. Even though it predicts with high accuracy if we include tilt faces if the face detected properly. Eventually, the total number of images equals 420 images. As we mentioned earlier, we consider $\pm 15^{\circ}$ as a frontal face. In this test, we are only trying to detect profile faces although some of the images also contain frontal faces, in total 420 images 150 for both sides and 120 for frontal face pose. The results of running the algorithm profile detector on the test images are shown in figure 9. The results on the HPID dataset achieve 91.19% accuracy and 91% precision, 90% recall, and 90% f1-score on selected set (consisting of only faces with $0^{\circ}$). Some example detections are shown in figure 10 ROC curve and confusion matrix. We wanted to confirm that the test set has only a faces with a tilt equal to $0^{\circ}$.




(a) The Confusion Matrix

(b) The ROC curves on the HPID dataset.

Figure 9: The ROC and Confusion Matrix Results for HPID dataset.

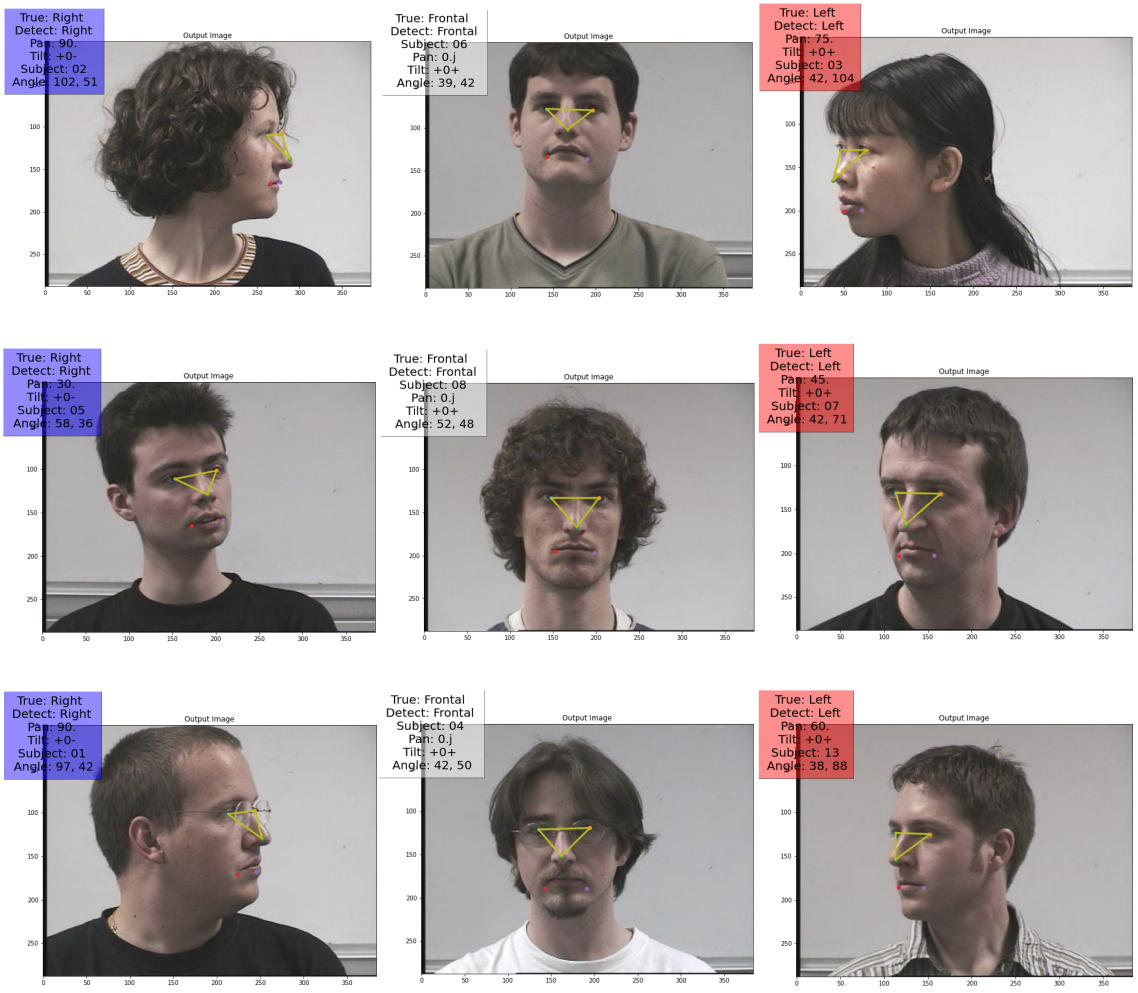
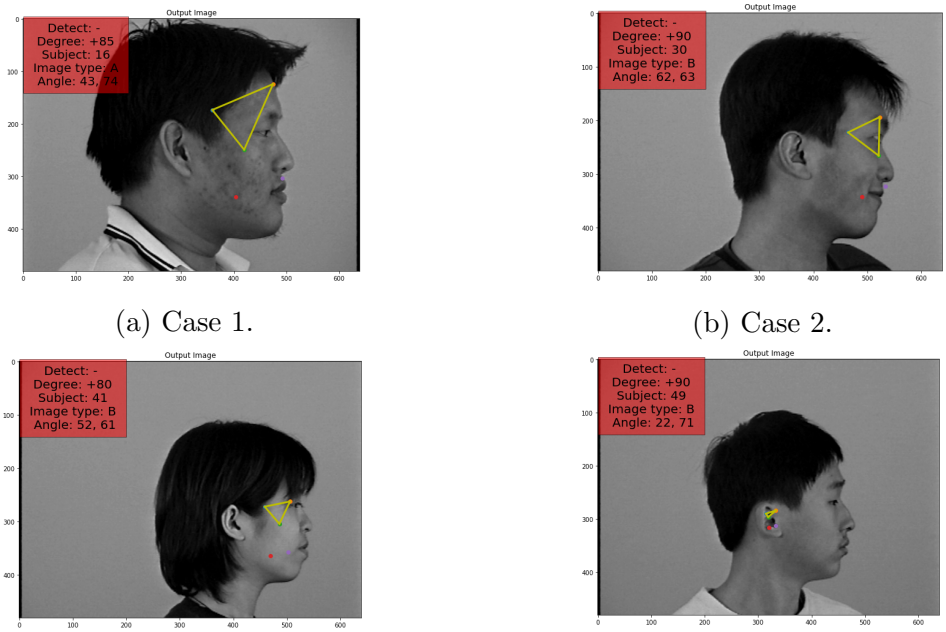



Figure 10: Example Predictions for Face Pose.

# 6 Discussion

We believe that our approach landmark-to-pose method, might have limitations for example:

- It is dependent on the face and landmark detection model.
- An additional problem is caused by the model. Which is hair is the part when the subject is in one of the side profiles, hair is more visible in the image and that cause confusion to the model and it might predict the ear or part of the face as a face as shown in figures 11a, 11b, 11c, and 11d.
- We only output three pose predictions which is not the case for the latter method when the landmark detection method fails.




(a) Case 1.

(b) Case 2.

(c) Case 3.

(d) Case 4.

Figure 11: False Detections.

However, it has the potential to be much simpler, more accurate, and faster against DL approaches. On the other hand, we cannot accurately determine whether a face is a side profile, left profile, or frontal face, and this is arguably one of the most difficulties in this area of research since it affects the labeling process. As shown

in figure 12 some of the false predictions because of our point of view with what is frontal face.

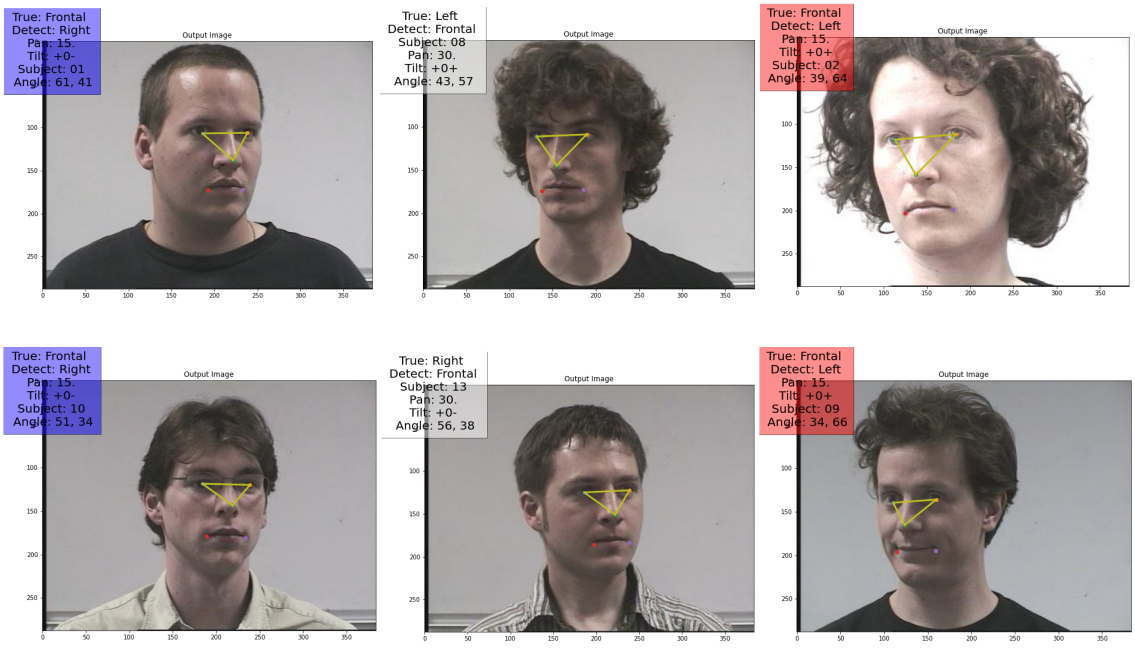



Figure 12: False Prediction on Face Pose.

# 7 Conclusion

This work has introduced an approach for the problem of face pose prediction using detection model as a core solution, and by calculating the angles between three face landmarks, then using a predefined threshold on produced angles to predict the face pose, either frontal pose, right or left side pose. The performance of the approach and threshold on pose discrimination unprecedented accuracy even on among recent work. The approach archived 91.19% on Head Pose Image Database. We show that our method shows robustness in cases of good detection. Future work will evaluate more our algorithm on a different dataset. Also, for applications that require a fully automated system, our algorithm may be combined to introduce additional feature face pose detection.

# References


[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[2] M. Jones and P. Viola, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-20003-96*, vol. 3, no. 14, p. 2, 2003.

[3] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2879–2886.

[4] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[5] F. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni, "Faceposenet: Making a case for landmark-free face alignment," 2017.

[6] R. Lab, "ourdatabase," http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm#Our_Database_, 2019.

[7] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," UMass Amherst technical report, Tech. Rep., 2010.

[8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[9] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[11] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial features," in *ICPR International Workshop on Visual Observation of Deictic Gestures*. Citeseer, 2004.