Evaluation of Facial Landmark Detection on MOBIO Database

Na Zhang

Problem

- MOBIO [1] database was usually used for…before?
- Seldom used in facial landmark detection
- In our problem,
  - Choose several state-of-art facial landmark detection methods
  - Execute landmark detection on MOBIO database
  - Evaluate the performance of popular detection methods on MOBIO

[*] Chris McCool, Sébastien Marcel, Abdenour Hadid, Matti Pietikäinen, Pavel Matějka, Jan Černocký, Norman Poh, Josef Kittler, Anthony Larcher, Christophe Lévy, Driss Matrouf, Jean-François Bonastre, Phil Tresadern, and Timothy Cootes, "Bi-Modal Person Recognition on a Mobile Phone: using mobile phone data", in IEEE ICME Workshop on Hot Topics in Mobile Mutlimedia, 2012.

MOBIO Database Description

- Mobile Biometrics Database
- Diverse Bi-modal database
- Consists of bi-modal data
  - Audio
  - Video
- Taken from 152 people
- Female-Male ratio: 1:2
  - 100 males
  - 52 females
- Collected from August 2008 until July 2010 in six different sites from five different countries with both native and non-native English speakers
- Source download link: https://www.idiap.ch/dataset/mobio

- 12 sessions were captured for each client
  - 6 sessions for Phase I
    - Consists of 21 questions with the question types ranging from:
    - Short Response Questions, Short Response Free Speech, Set Speech, and Free Speech
  - 6 sessions for Phase II
    - Consists of 11 questions with the question types ranging from:
    - Short Response Questions, Set Speech, and Free Speech
- Recorded using 2 mobile devices
  - A mobile phone: NOKIA N93i
  - A laptop computer: standard 2008 MacBook
- The laptop was only used to capture part of the first session
- The first session consists of data captured on both the laptop and the mobile phone

Detailed Description of Questions

- Short Response Questions

  The short response questions consisted of five pre-defined questions, which were:

  - What is your name? – the user supplies their fake name
  - What is your address? – the user supplies their fake address
  - What is your birthdate? – the user supplies their fake birthdate
  - What is your license number? – the user supplied their fake ID card number (the same for each person)
  - What is your credit card number? – the user supplies their fake Card number

- Short Response Free Speech
  - There were five random questions taken form a list of 30-40 questions.
  - The user had to answer these questions by speaking for approximately 5 seconds of recording (sometimes more and sometimes less).

- Set Speech
  - The users were asked to read pre-defined text out aloud
  - This text was designed to take longer than 10 seconds to utter and the participants were allowed to correct themselves while reading these paragraphs.

- Free Speech
  - Consisted of 10 random questions from a list of approximately 30 questions
  - The answers to each of these questions took approximately 10 seconds (sometimes less and sometimes more)

- In our problem:
  - Extract frames from video data
  - Just collect still face images
  - 20,600 face images with 640*480 size

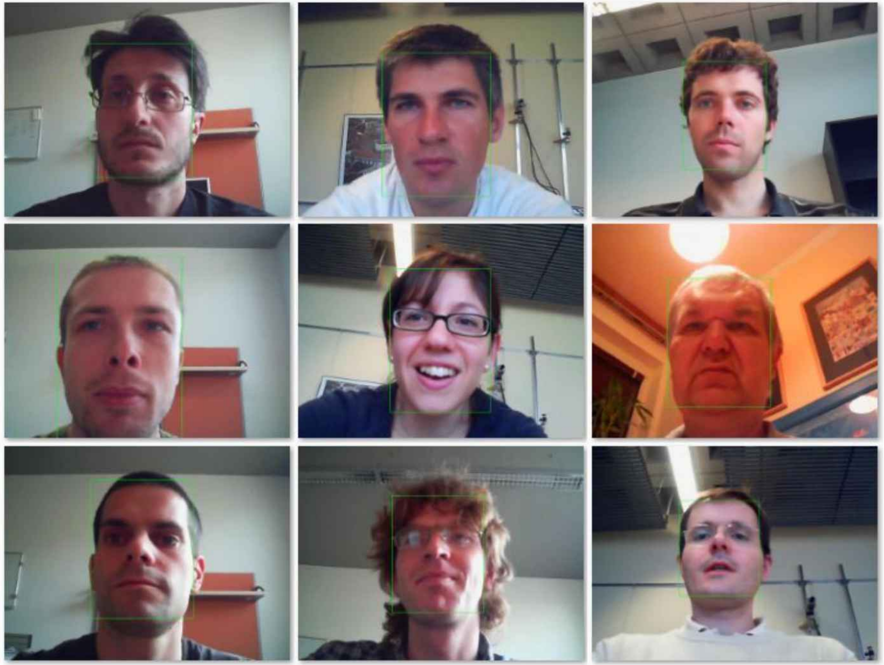
Preprocess Data

- Face Detection
    - MTCNN [*] vs. MatLab Dlib
    - MatLab Dlib:
        - 18, 483 images with one correct face
        - 537 with multiple faces
        - 1,580 with no face

[*] Zhang, Kaipeng, et al. "Joint face detection and alignment using multitask cascaded convolutional networks." IEEE Signal Processing Letters 23.10 (2016): 1499-1503.

- MTCNN:
  - 20,275 images with one right face
  - 211 with multiple faces
  - 114 with no face
  - **Finally, 20,481 images detected**

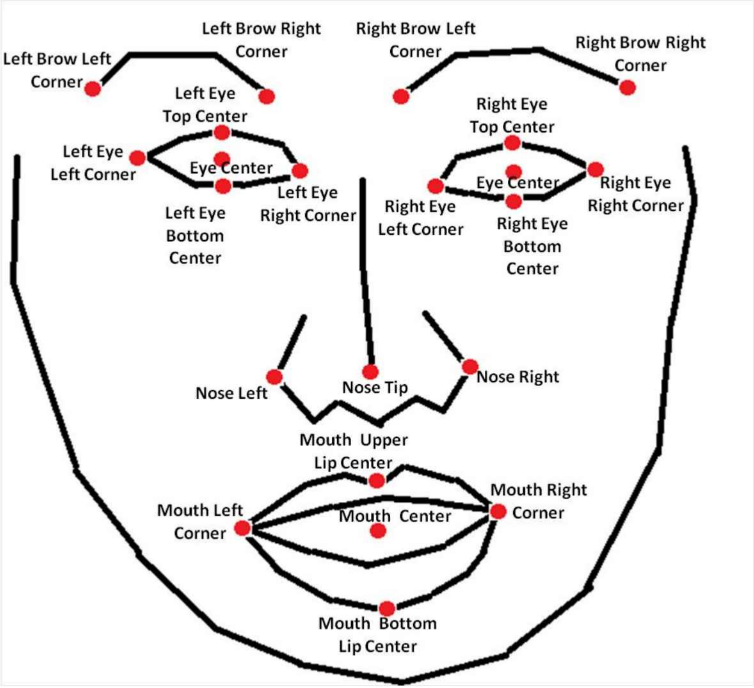
- Choose MTCNN!!!



- Face Cropping
  - Crop into square shape with fixed size by bounding box information
  - 300 * 300
- Face Resize
  - Different facial landmark detection methods have different input size
    - 256 * 256
    - 227 * 227
    - 224*224
    - …

Generate Ground Truth

- 1.Left brow left corner
- 2.Left brow right corner
- 3.Right brow left corner
- 4.Right brow right corner
- 5.Left eye left corner
- 6.left eye top center
- 7.Left eye right corner
- 8.left eye bottom center
- 9.left eye center
- 10.Right eye left corner
- 11.right eye top center
- 12.Right eye right corner
- 13.right eye bottom center
- 14.right eye center
- 15.Nose tip
- 16.Nose left
- 17.Nose right
- 18.Mouth left corner
- 19.mouth upper lip center
- 20.Mouth right corner
- 21.Mouth bottom lip center
- 22.Mouth center



- Manually label 22 facial landmarks
- During 2014 to 2017
- Develop a Labeling Tool
    - Named FaceLabel_App
    - The result saved in .txt files

Face Label App

- Run in windows system
- Label images one by one
- In order

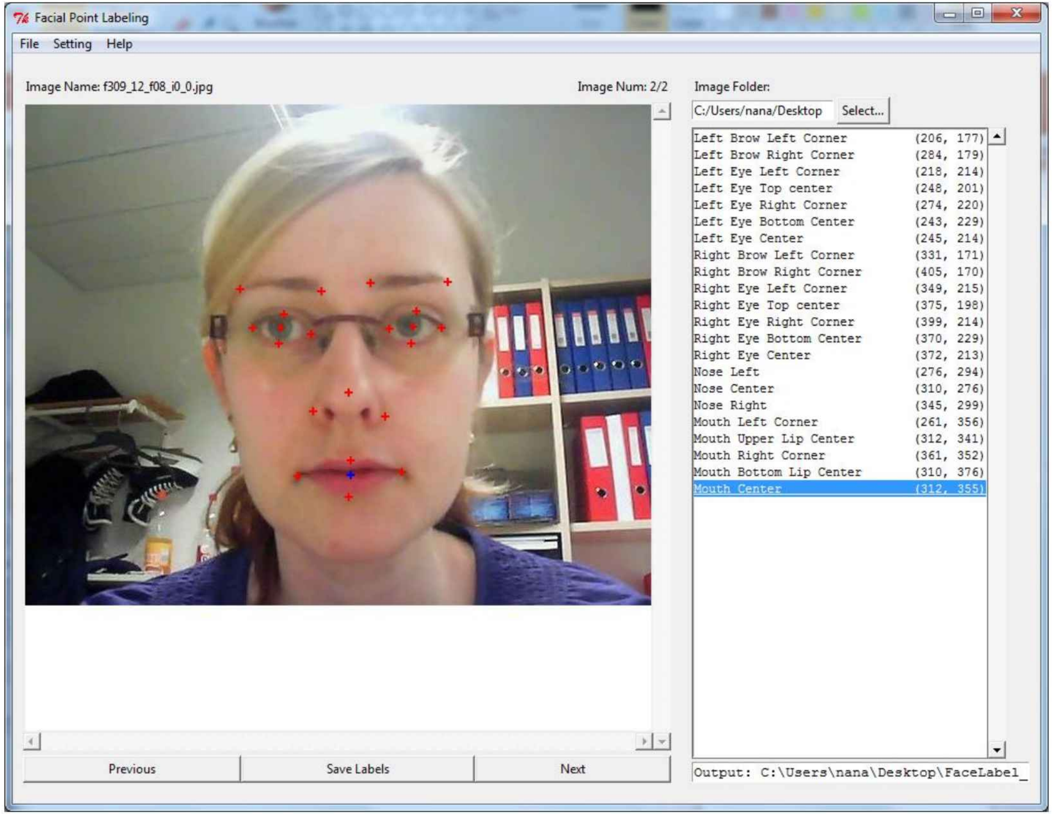


Image Name: f309_12_f08_i0_0.jpg

Image Num: 2/2

Image Folder: C:/Users/nana/Desktop

Left Brow Left Corner	(206, 177)
Left Brow Right Corner	(284, 179)
Left Eye Left Corner	(218, 214)
Left Eye Top center	(248, 201)
Left Eye Right Corner	(274, 220)
Left Eye Bottom Center	(243, 229)
Left Eye Center	(245, 214)
Right Brow Left Corner	(331, 171)
Right Brow Right Corner	(405, 170)
Right Eye Left Corner	(349, 215)
Right Eye Top center	(375, 198)
Right Eye Right Corner	(399, 214)
Right Eye Bottom Center	(370, 229)
Right Eye Center	(372, 213)
Nose Left	(276, 294)
Nose Center	(310, 276)
Nose Right	(345, 299)
Mouth Left Corner	(261, 356)
Mouth Upper Lip Center	(312, 341)
Mouth Right Corner	(361, 352)
Mouth Bottom Lip Center	(310, 376)
Mouth Center	(312, 355)

```
Output: C:\Users\nana\Desktop\FaceLabel_
```

Experiment & Evaluation

- Choose several facial landmark detection methods to detect landmarks
- Compare the points with ground truth for evaluation
- **Measure metric**
    - NME: Normalized Mean Error
    - CED: Cumulative Error Distribution Curve
    - AUC: Area Under the error Curve
    - Failure rate

Mean Normalized Error

- The Euclidean Distance (L_2 $norm$) between estimated points and ground truth are normalized by inter-ocular/ outer eye corner distance

$$e_i = \frac{\left\| X^e_{(i)} - X^g_{(i)} \right\|_2}{d_{io}}$$

e_i: the i-th error value
$X^e_{(i)}$: the i-th estimated points
$X^g_{(i)}$: the i-th ground truth
d_{io}: IOD, the inter-ocular distance, i.e. Euclidean distance between two eye centers

- NME can be:
  - Sample-wise
  - Landmark-wise, like above
  - Overall
- Heavy impacted by outliers

- Use the **distance of two outer eye corners from ground truth** to normalize
- Use landmark-wise NME
- For every face image:
  - Calculate Euclidean distance of 2 outer eye centers: d
  - Calculate the sum of Euclidean distances for 15/16/5 facial landmarks: $\sum_{i=1}^{15} D_i$
  - Calculate normalized mean error: $error = \frac{\sum_{i=1}^{15} D_i}{15 * d}$

Notes: [68 points: 15] ; [19 points: 16]; [5 points: 5]

Cumulative Error Distribution

- Cumulative distribution function of normalized errors
- Evaluate the fraction of facial landmarks changes as error threshold changes
- Better way to handle outliers
- In our experiment,
    - We set error value threshold is 0.08
    - Partition the error value range [0, 0.08] into 80 segments with equal step size 0.001
    - For each error value point X, Calculate the fraction of face images whose error value <= X as Y

AUC

- The area under the error curve CED

$$AUC_{\alpha} = \int_{0}^{\alpha} f(e)de$$

e: Normalized error

f(e):cumulative error distribution function

α:upper bound, used to calculate the define integration

Failure Rate

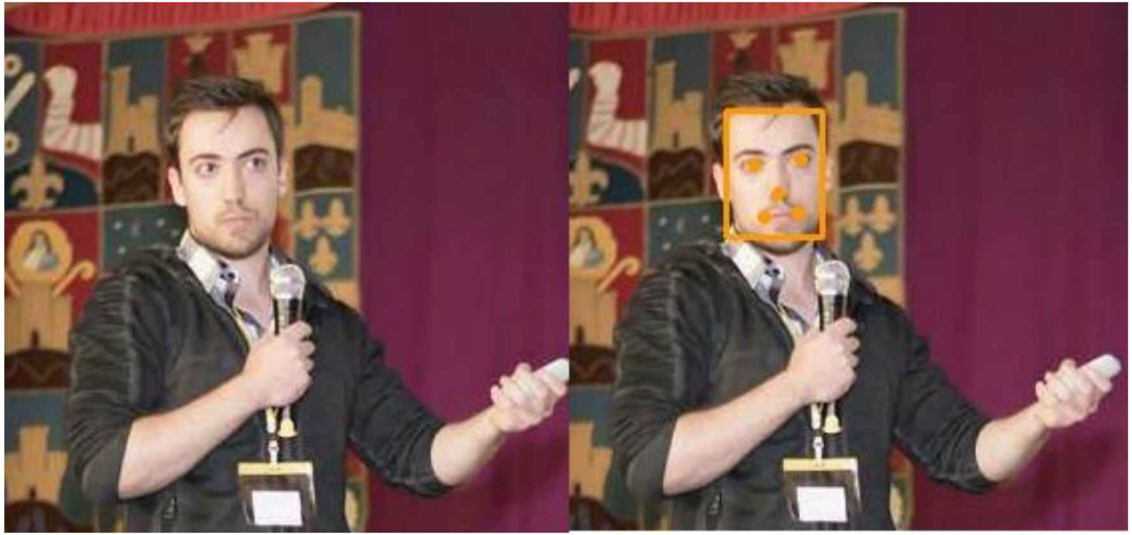
- Count the fraction of faces whose error value is greater than error value threshold, e.g. 0.08

Facial Landmark Detection Methods

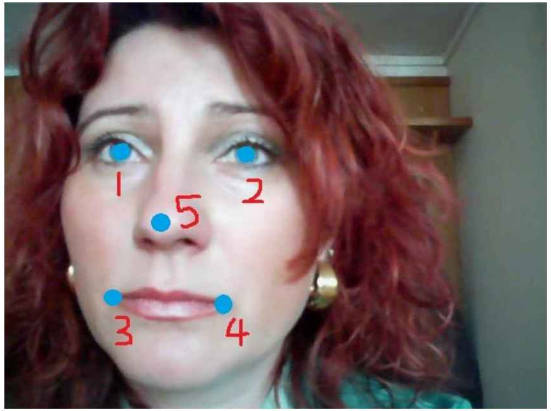
- Tweaked CNN
- WingLoss
- DAC-CSR
- PA-CNN
- OpenPose
- ECT
- TCDCN

MTCNN

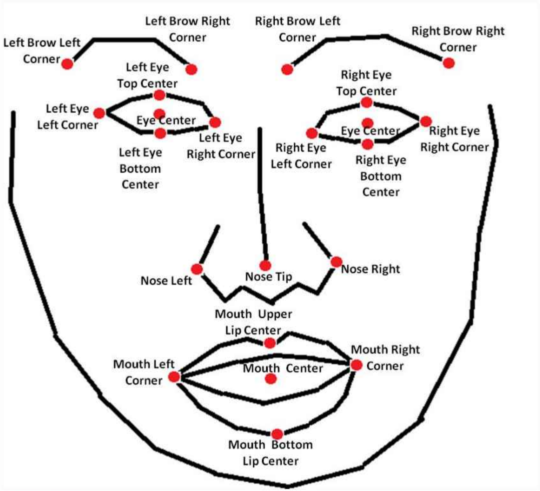
- Python3.0 + mtcnn
- 18,392
- 5 points
- Input original images

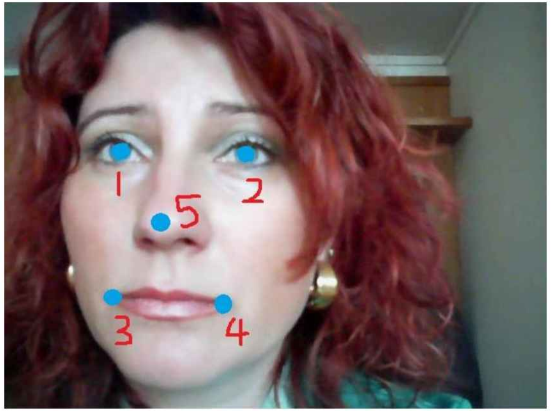


- 1. left eye center
- 2. right eye center
- 3. mouth left corner
- 4. mouth right corner
- 5. nose tip



- For 5 detected landmarks:
  - Find the facial points that can be get their corresponding points in those 22 ground truth points for evaluation
- Find 5 points in total
  - 9.left eye center -- 1
  - 14.right eye center -- 2
  - 18.Mouth left corner -- 3
  - 20.Mouth right corner – 4
  - 15.Nose tip -- 5



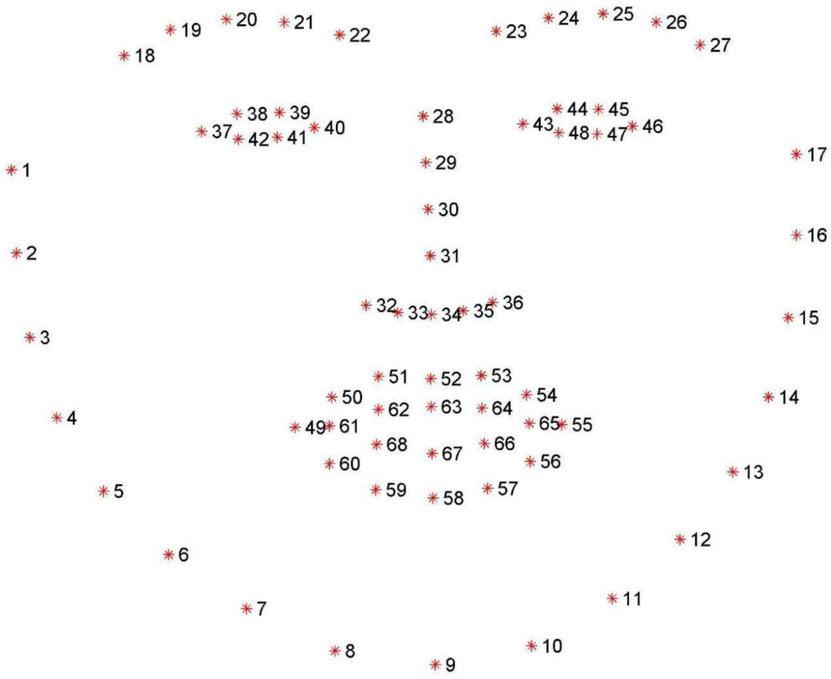


ECT Model

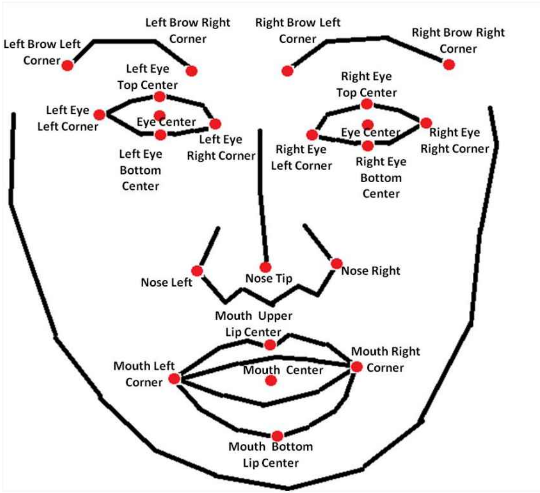
- Estimation Correction Tuning Deep Model
  - Data-driven model: FCN; compute response maps (textural appearance information)
  - Model-driven model: Maximum points fitted with PDM
  - RLMS: fine-tune facial shape iteratively, correct outliers of landmarks
- Pre-trained deep model
- On Caffe + Python
- Input: 256 * 256
- Output: 68 facial points

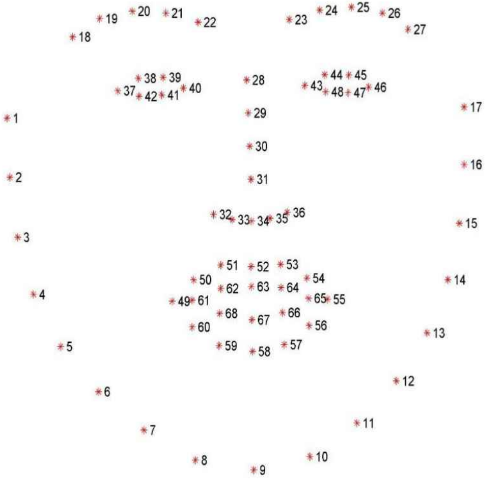
68 facial points

- 51 facial features points
  - 5+5 brow
  - 6+6 eyes
  - 9 nose
  - 20 mouth
- 17 face contour points

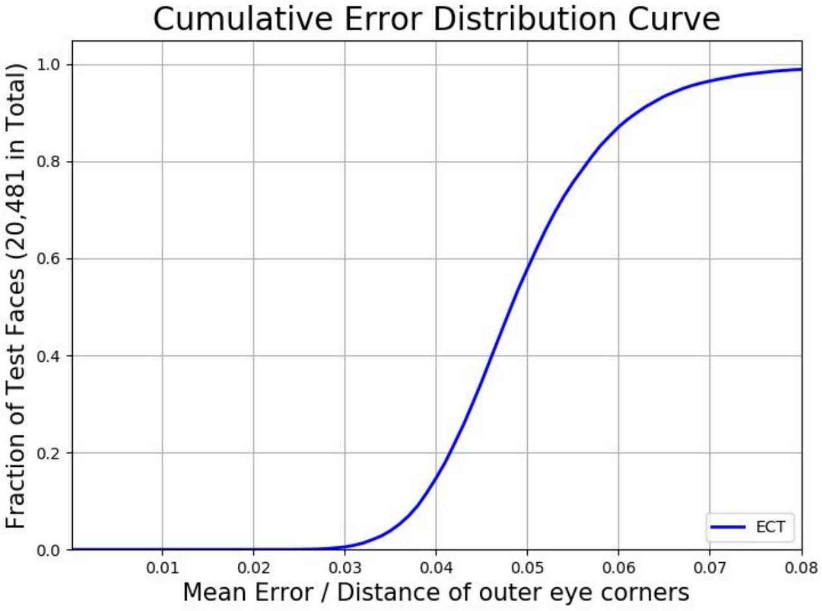


- For 68 detected landmarks:
  - Find the facial points that can be get their corresponding points in those 22 ground truth points for evaluation
- Find 15 points in total
  - 1.Left brow left corner -- 18
  - 2.Left brow right corner -- 22
  - 3.Right brow left corner -- 23
  - 4.Right brow right corner -- 27
  - 5.Left eye left corner -- 37
  - 7.Left eye right corner -- 40
  - 10.Right eye left corner -- 43
  - 12.Right eye right corner -- 46
  - 15.Nose tip -- 31
  - 16.Nose left -- 32
  - 17.Nose right -- 36
  - 18.Mouth left corner -- 49
  - 20.Mouth right corner -- 55
  - 19.Mouth upper lip center -- 52
  - 21.Mouth bottom lip center -- 58





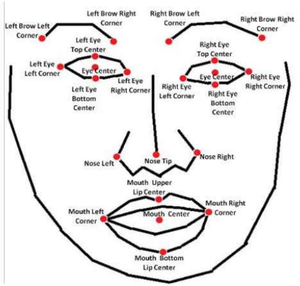
- 20,481 normalized mean errors
- Set error threshold=0.08
- Step size=0.001
- AUC=38.226405
- Failure rate: 1.079049%

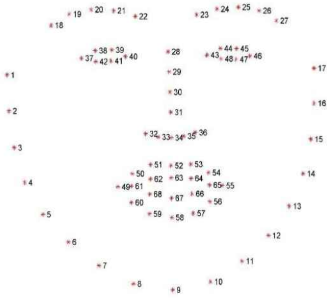


PA-CNN

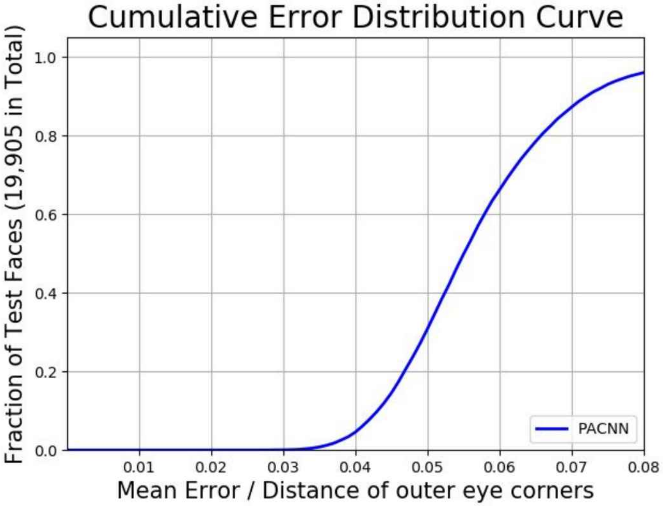
- Part-Aware Deep CNN
- End-to-end regression framework
  - Encode image into feature maps shared by all landmarks
  - The feature are sent into 2 sub-nets to regress 2 types of landmarks
    - Contour landmark: 17
    - Inner landmark: 51
  - **Can directly detect landmarks on original images**
  - **Does not need to detect, crop, and resize face**
- Caffe + Python + Dlib + OpenCV3
- Output: 68 points

- In total
  - 19,505 faces are detected
  - 1,095 faces fail to be detected
- Evaluation is similar with Method 1
- Adopt same 15 landmarks for evaluation





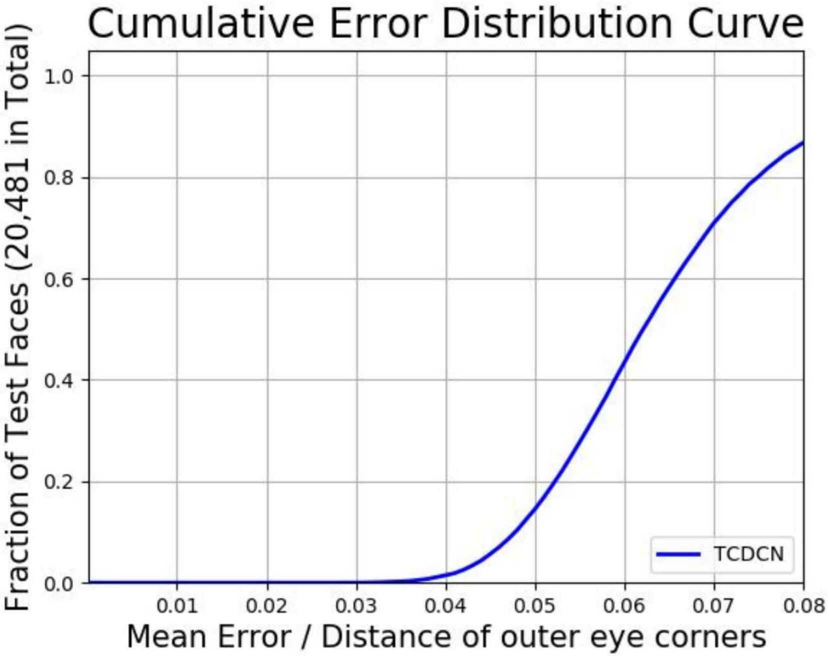
- 19,905 normalized mean errors
- Set error threshold=0.08
- Step size=0.001
- AUC=29.574564
- Failure rate: 4.029736%



TCDCN

- 68 points:
- input: original images,
- bbox: [left, top, width, height]
- output:
  - 20,481 images:
  - 68 facial landmark: (x1,y1,x2,y2....x68,y68).
- Evaluation is similar with Method 1
- Adopt same 15 landmarks for evaluation

- 20,481 normalized mean errors
- Set error threshold=0.08
- Step size=0.001
- AUC (%) = 21.545304
- Failure Rate (%) = 13.290367

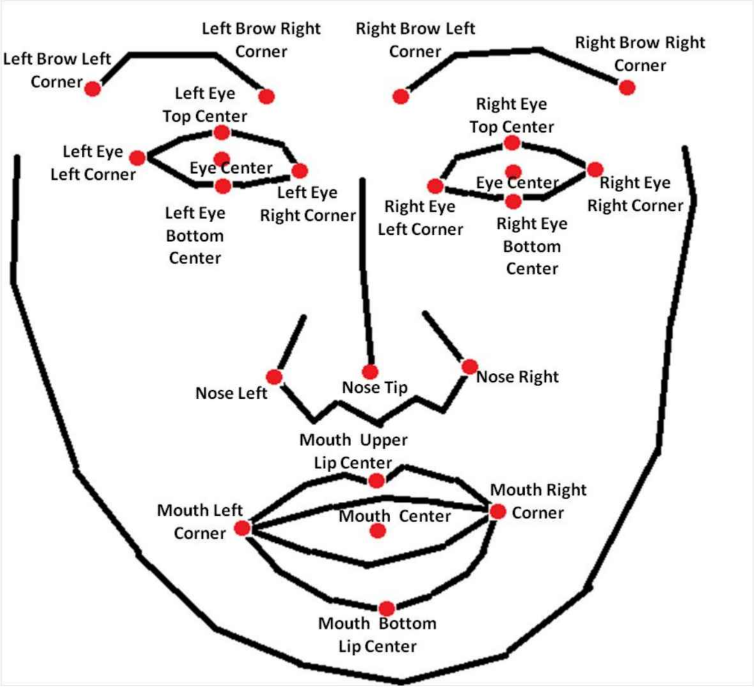


WingLoss

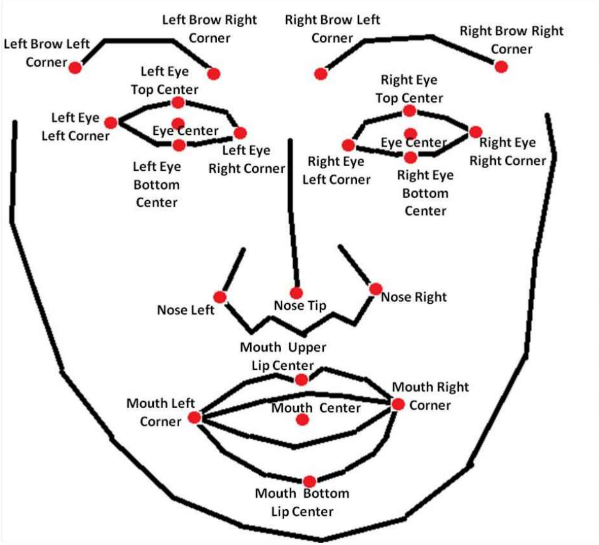
- input: in 256*256 size;
- MTCNN face detection
- output: 19 landmarks.
- (x1,x2,x3,....x19, y1,y2,y3,.....y19)

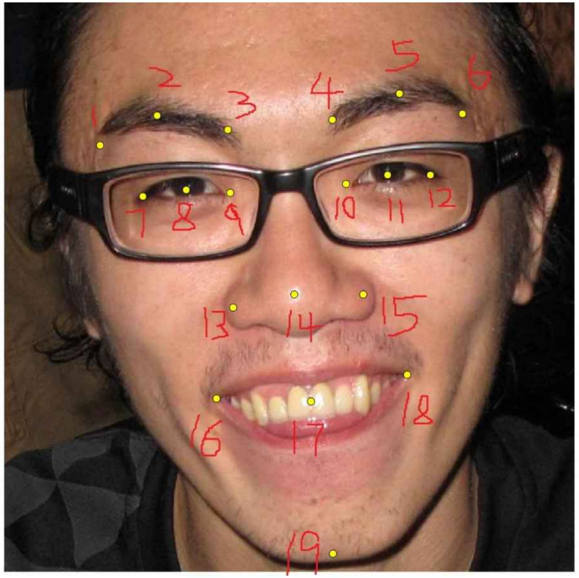
Ground Truth

- 1.Left brow left corner
- 2.Left brow right corner
- 3.Right brow left corner
- 4.Right brow right corner
- 5.Left eye left corner
- 6.left eye top center
- 7.Left eye right corner
- 8.left eye bottom center
- 9.left eye center
- 10.Right eye left corner
- 11.right eye top center
- 12.Right eye right corner
- 13.right eye bottom center
- 14.right eye center
- 15.Nose tip
- 16.Nose left
- 17.Nose right
- 18.Mouth left corner
- 19.mouth upper lip center
- 20.Mouth right corner
- 21.Mouth bottom lip center
- 22.Mouth center



- For 19 detected landmarks:
  - Find the facial points that can be get their corresponding points in those 22 ground truth points for evaluation
- Find 16 points in total
  - 1.Left brow left corner -- 1
  - 2.Left brow right corner -- 3
  - 3.Right brow left corner -- 4
  - 4.Right brow right corner -- 6
  - 5.Left eye left corner -- 7
  - 9.left eye center -- 8
  - 7.Left eye right corner -- 9
  - 10.Right eye left corner -- 10
  - 14.right eye center -- 11
  - 12.Right eye right corner -- 12
  - 16.Nose left -- 13
  - 15.Nose tip -- 14
  - 17.Nose right -- 15
  - 18.Mouth left corner -- 16
  - 22.Mouth center -- 17
  - 20.Mouth right corner -- 18





- WingLoss: 1,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,

DAC-CSR

- img_list.txt:

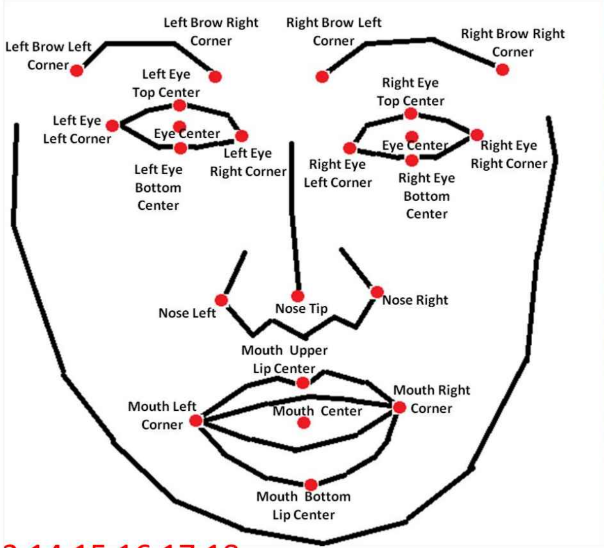
- record source image path and bbox [x1, y1, width, height].

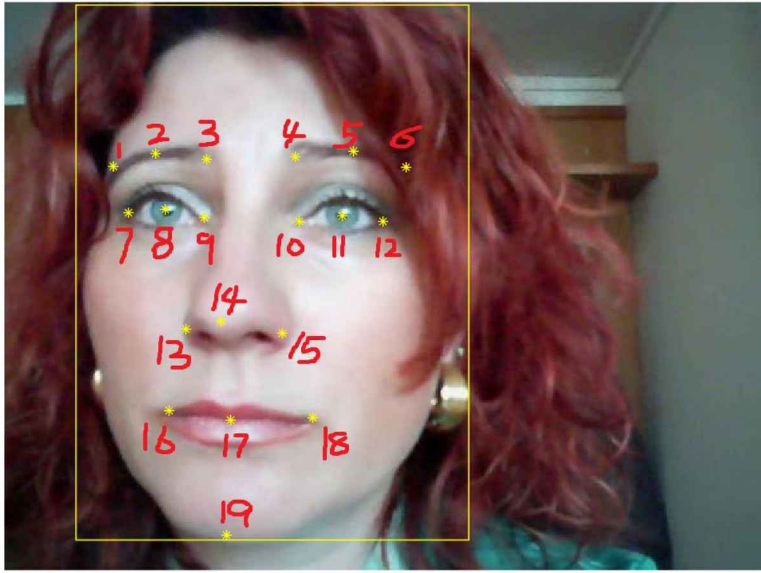
- input:
- Original images, 20481 images

- MTCNN bbox
- output: 19 landmarks

- [x1, x2,...,x19, y1, y2,.....y19]

- For 19 detected landmarks:
  - Find the facial points that can be get their corresponding points in those 22 ground truth points for evaluation
- Find 16 points in total
  - 1.Left brow left corner -- 1
  - 2.Left brow right corner -- 3
  - 3.Right brow left corner -- 4
  - 4.Right brow right corner -- 6
  - 5.Left eye left corner -- 7
  - 9.left eye center -- 8
  - 7.Left eye right corner -- 9
  - 10.Right eye left corner -- 10
  - 14.right eye center -- 11
  - 12.Right eye right corner -- 12
  - 16.Nose left -- 13
  - 15.Nose tip -- 14
  - 17.Nose right -- 15
  - 18.Mouth left corner -- 16
  - 22.Mouth center -- 17
  - 20.Mouth right corner -- 18





- WingLoss: 1,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,

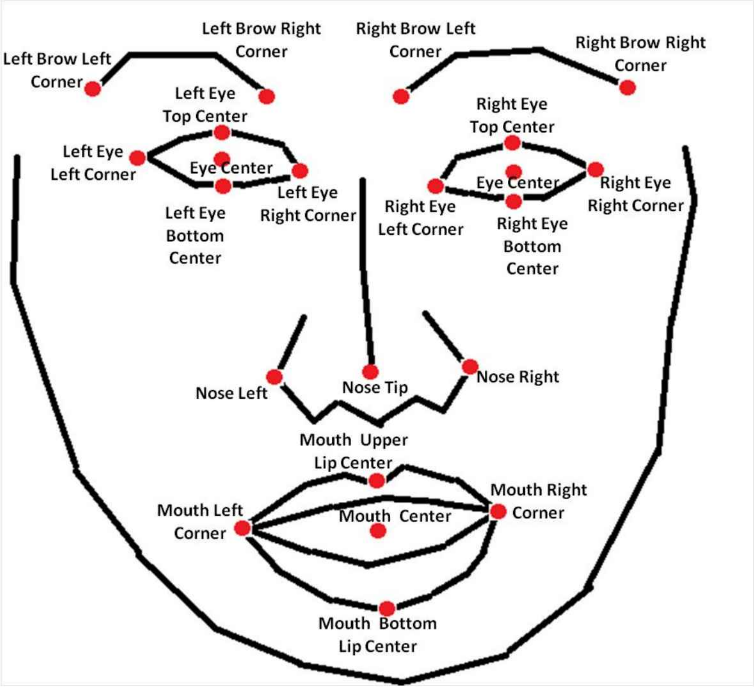
VillianCNN

- 1: left eye center
- 2: right eye center
- 3: nose tip
- 4: left mouth corner
- 5: right mouth corner

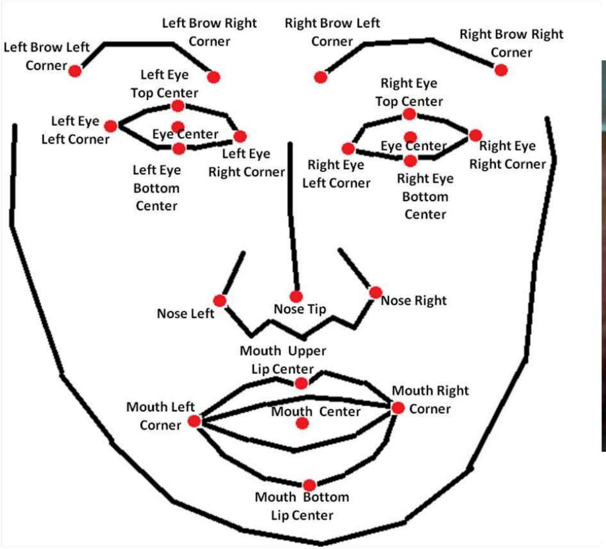
- input any size images,
- -- MOBIO Faces, we input 256*256!
- 20,481 images in total.
- resize to 40*40
- output: 40*40

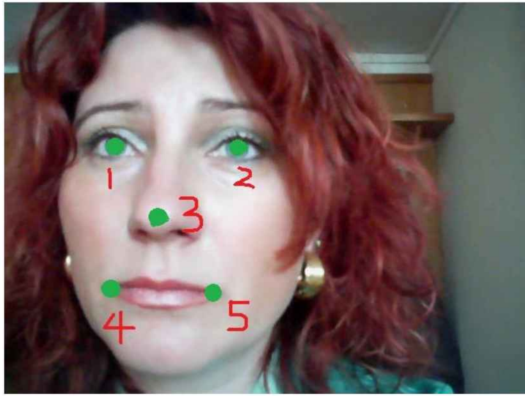
Ground Truth

- 1.Left brow left corner
- 2.Left brow right corner
- 3.Right brow left corner
- 4.Right brow right corner
- 5.Left eye left corner
- 6.left eye top center
- 7.Left eye right corner
- 8.left eye bottom center
- 9.left eye center
- 10.Right eye left corner
- 11.right eye top center
- 12.Right eye right corner
- 13.right eye bottom center
- 14.right eye center
- 15.Nose tip
- 16.Nose left
- 17.Nose right
- 18.Mouth left corner
- 19.mouth upper lip center
- 20.Mouth right corner
- 21.Mouth bottom lip center
- 22.Mouth center



- For 5 detected landmarks:
  - Find the facial points that can be get their corresponding points in those 22 ground truth points for evaluation
- Find 5 points in total
  - 9.left eye center -- 1
  - 14.right eye center -- 2
  - 15.Nose tip -- 3
  - 18.Mouth left corner -- 4
  - 20.Mouth right corner -- 5
- VillianCNN: 1,2,3,4,5





Final Result Comparison

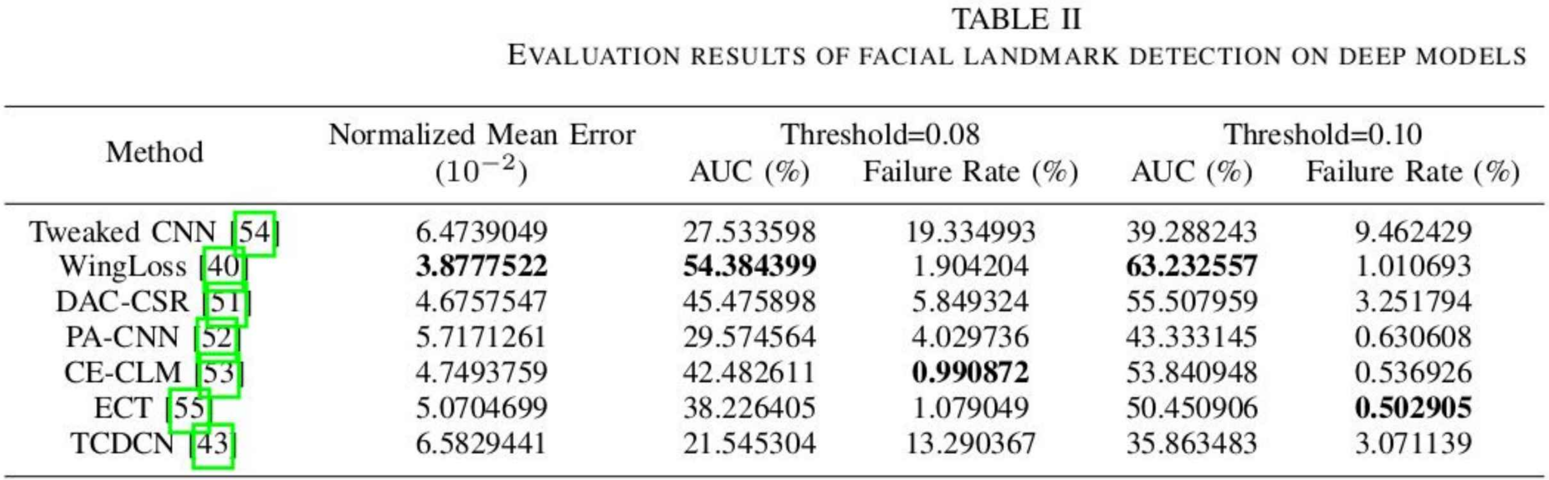
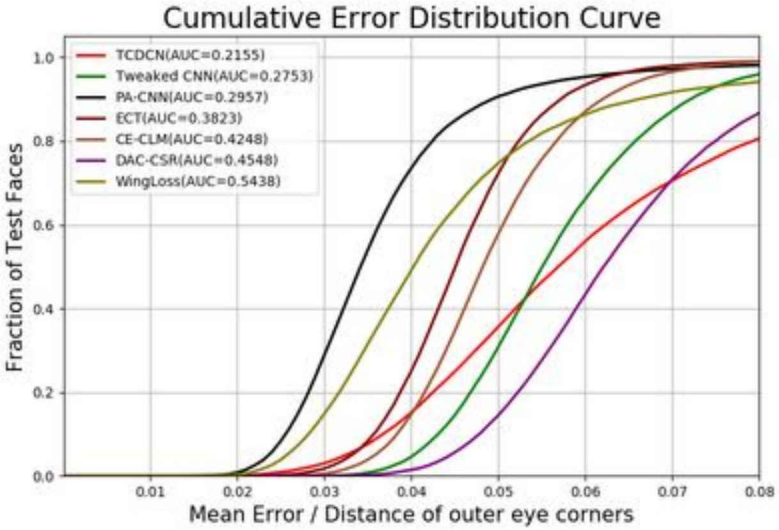
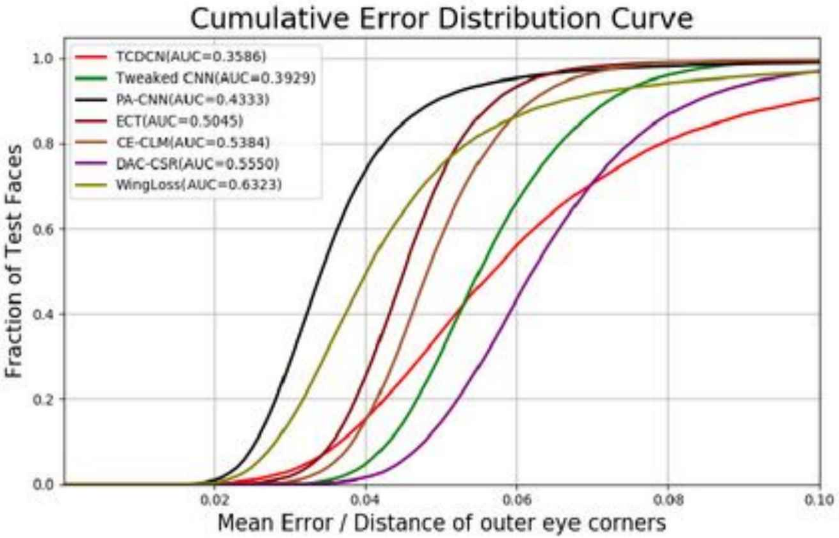


TABLE II
EVALUATION RESULTS OF FACIAL LANDMARK DETECTION ON DEEP MODELS

Method	Normalized Mean Error	Threshold=0.08		Threshold=0.10	
	(10^{-2})	AUC (%)	Failure Rate (%)	AUC (%)	Failure Rate (%)
Tweaked CNN [54]	6.4739049	27.533598	19.334993	39.288243	9.462429
WingLoss [40]	**3.8777522**	**54.384399**	1.904204	**63.232557**	1.010693
DAC-CSR [51]	4.6757547	45.475898	5.849324	55.507959	3.251794
PA-CNN [52]	5.7171261	29.574564	4.029736	43.333145	0.630608
CE-CLM [53]	4.7493759	42.482611	**0.990872**	53.840948	0.536926
ECT [55]	5.0704699	38.226405	1.079049	50.450906	**0.502905**
TCDCN [43]	6.5829441	21.545304	13.290367	35.863483	3.071139


Cumulative Error Distribution Curve
TCDCN(AUC=0.2155)
Tweaked CNN(AUC=0.2753)
PA-CNN(AUC=0.2957)
ECT(AUC=0.3823)
CE-CLM(AUC=0.4248)
DAC-CSR(AUC=0.4548)
WingLoss(AUC=0.5438)
Fraction of Test Faces
0.0
0.2
0.4
0.6
0.8
1.0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Mean Error / Distance of outer eye corners

(a)


Cumulative Error Distribution Curve
TCDCN(AUC=0.3586)
Tweaked CNN(AUC=0.3929)
PA-CNN(AUC=0.4333)
ECT(AUC=0.5045)
CE-CLM(AUC=0.5384)
DAC-CSR(AUC=0.5550)
WingLoss(AUC=0.6323)
Fraction of Test Faces
0.0
0.2
0.4
0.6
0.8
1.0
0.02
0.04
0.06
0.08
0.10
Mean Error / Distance of outer eye corners

(b)