Face Detection

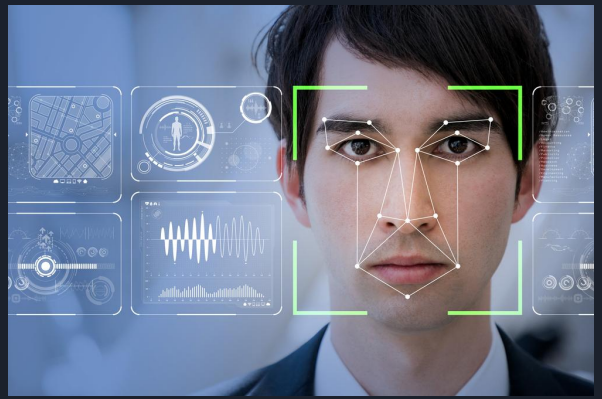
John Mikos and Amol Sayala

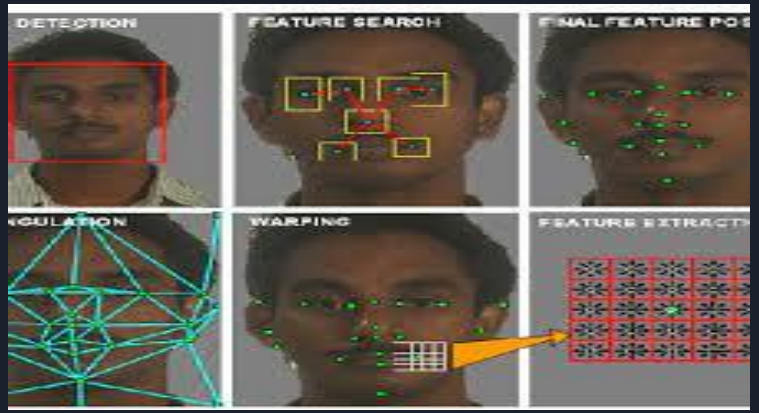
What is Face detection?

https://www.kairos.com/demos

Google Def: is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene.

Simplified definition: is an artificial intelligence (AI) based computer technology used to find and identify human faces in digital images.

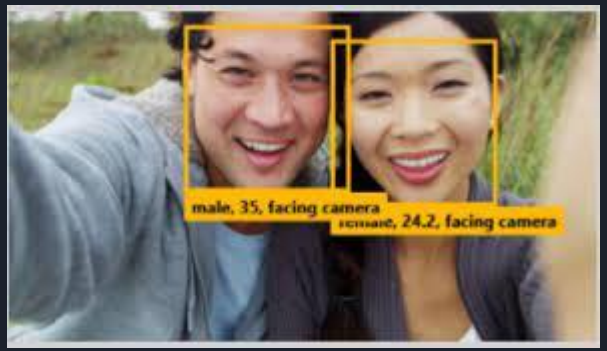




How is it used in real life?

Face detection has numerous applications, including people-counting, online marketing, and even the auto-focus of a camera lens

Example:



What are the pros and cons for face detection?

Pros

- Finding missing people and identifying perpetrators
- Protecting businesses against theft
- Better security measures in banks and airports
- Better analyze a person and their data for what they like
- Better tools for organising photos
- Better medical treatment

Cons:

- Greater threat to individual and societal privacy
- Can infringe on personal freedoms
- Violates your human rights
- Provides opportunities for fraud and other crimes
- Its not going to gather all the specific data that you need at times
- Technology can be fooled

https://www.itpro.com/security/privacy/356882/the-pros-and-cons-of-facial-recognition-technology

Robust Real-Time Face Detection

2003

Abstract Summarized

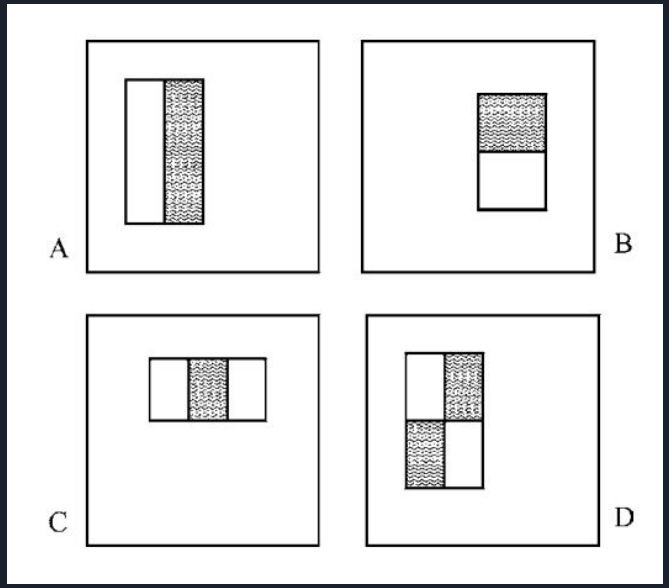
The deep neural networks built using this framework are designed to be light-weight and feature a streamlined architecture. We introduce two global hyperparameters that trade off between accuracy and latency.

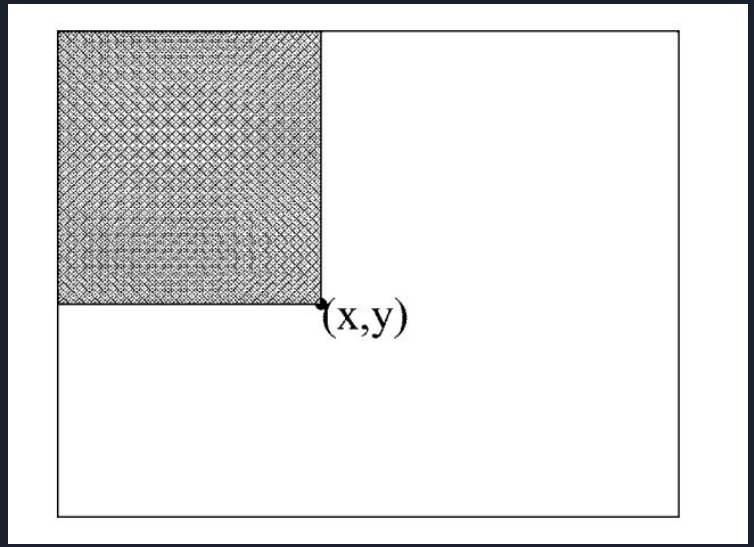
The deep neural networks built using this framework are designed to be light-weight and feature a streamlined architecture. We introduce two global hyperparameters that trade off between accuracy and latency.

Through two insights, we were able to achieve the best performance in the previous iteration of PASCAL VOC. The first is to implement high-capacity deep neural networks in bottom-up region proposals to perform segmentation and localization.

$$ii(x, y) = \sum_{x' \le x, y' \le y} i(x', y'),$$

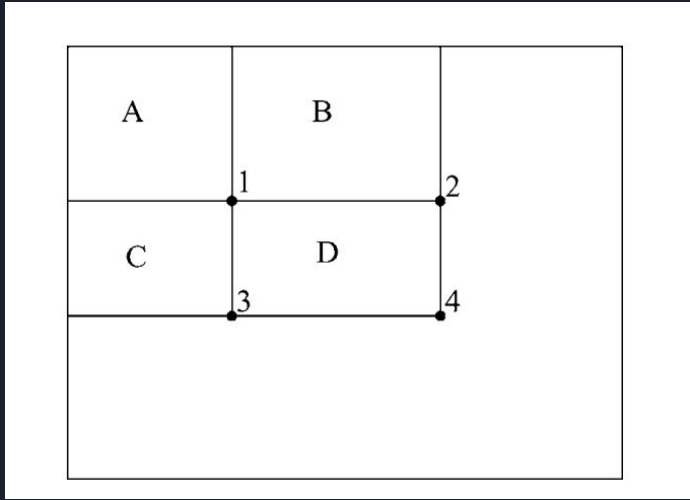
Features in motion for face detection





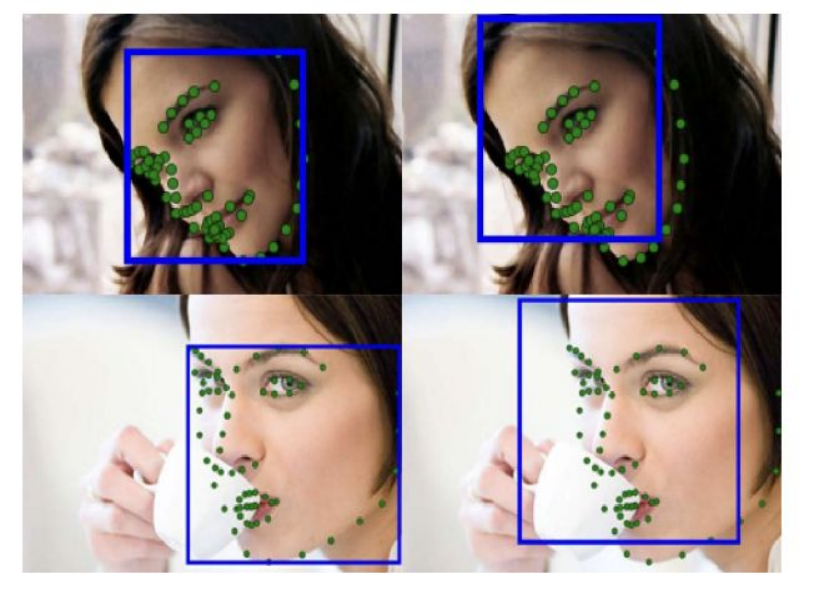
The value of the integral image at point (x, y) is the sum of all the pixels above and to the left.

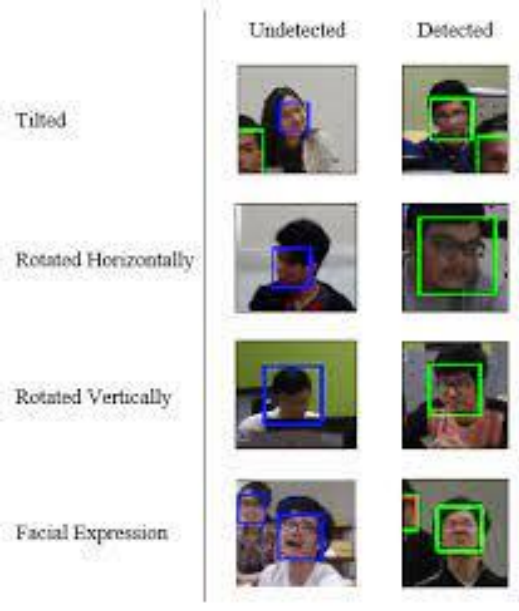
Internal Imaging



The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D. The sum within D can be computed as 4 + 1 − (2 + 3).

Focus on the green dots and squares





Steps overview

1. Training dataset
2. Structure of the Detector cascade
3. Speed of the final detector
4. Image Processing
5. Scanning the detector
6. Intergation of Multiple Detections
7. Experiements on the real world set

Training dataset

What data they take:

- The face shape
- The eyes
- Nose
- Defects
- Race
- Glasses (analyze how they look without them)
- The color of the photo (color correct)
- The angle

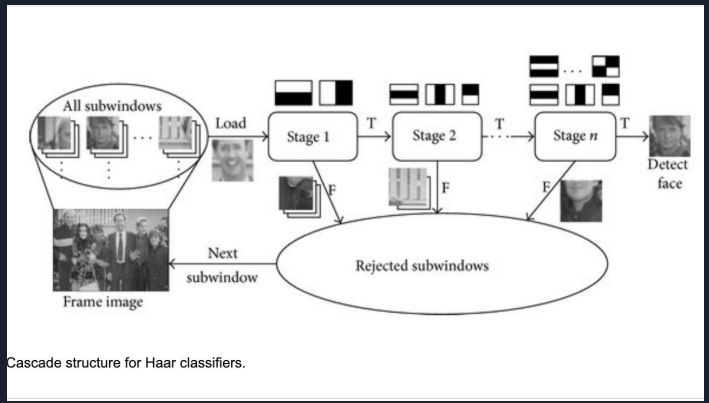
Structure of the detector Cascade

The final detector is a 38 layer cascade of classifiers

which included a total of 6060 features.

The first classifier in the cascade is constructed using two features and rejects about 50% of non-faces

while correctly detecting close to 100% of faces



Cascade structure for Haar classifiers.

Speed of the final Detector

	This work	SOVC2005[1]	VLSID2004[2]	CVPRW2004[3]	ICCE2003[4]
Function	Face / Object Detection	Image Filtering Processor for Face Detection	Face Detection	Face Recognition	Face Detection
Technology	0.13μm CMOS	0.35μm CMOS	0.16μm	0.13μm CMOS	0.18μm CMOS
Clock Frequency	100MHz	-	75.75MHz	150MHz	54MHz
Area	0.79mm² (CORE)	9.8mm²	30.4mm²	48.7mm²	76mm²
Speed	30fps	-	424fps	20fps	14fps
Power	29mW @ 1.2V	280mW @ 3.3V	7.35W @ 1.8V	1W @ 1.5V	90mW @ 1.8V

Image processing

We have to take in fact for the aspects for the face (white dots) and have that in data to then compare to other faces for similarity

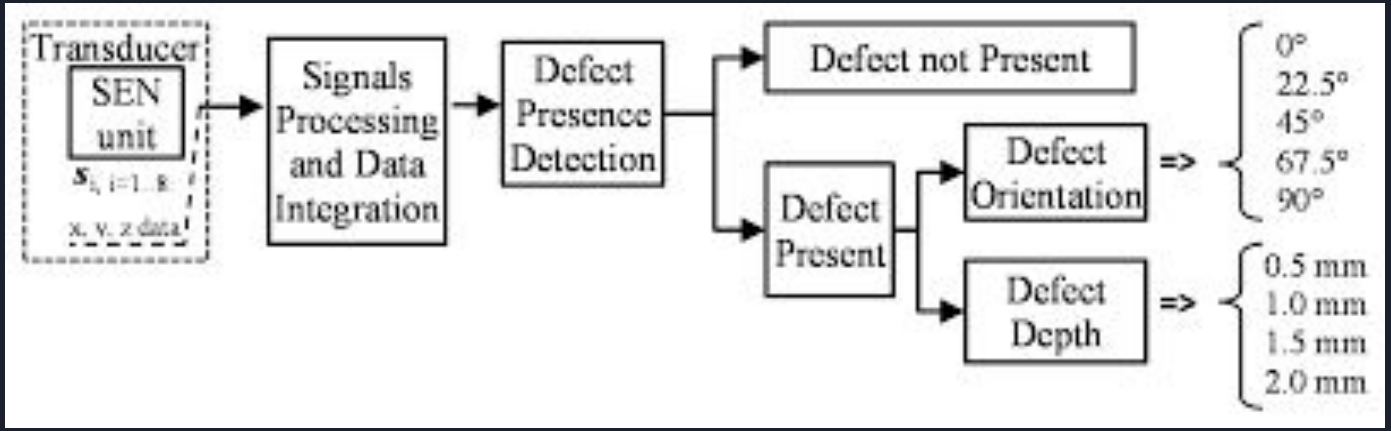
You also have to process the image for in case of fake face identity

Scanning the detector

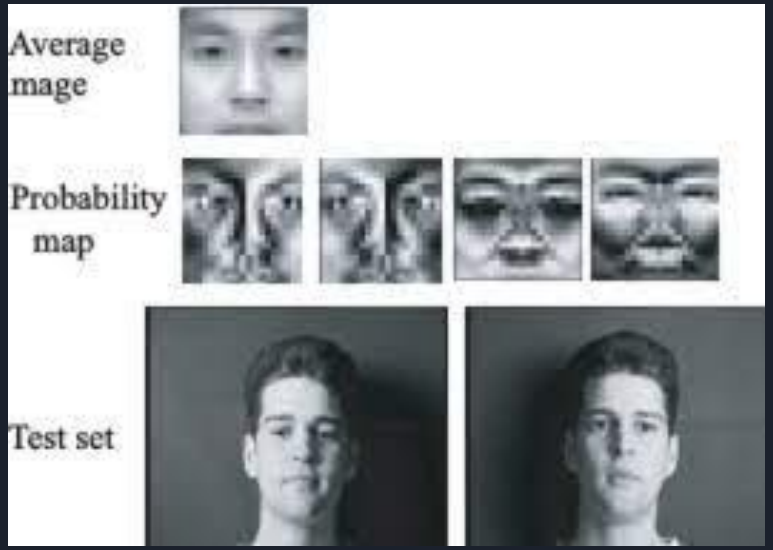
- The speed of the cascaded detector is directly related to the number of features evaluated per scanned subwindow.
- the number of features evaluated depends on the images being scanned.
- Since a large majority of the sub-windows are discarded by the first two stages of the cascade, an average of 8 features out of a total of 6060 are evaluated per sub-window
- This is roughly 15 times faster than the Rowley-BalujaKanade detector

Integration of Multiple detections

The process has a way to then determine the way of the photo and have that for data for future faces



Experiments on the real world set



Results

Table 3. Detection rates for various numbers of false positives on the MIT + CMU test set containing 130 images and 507 faces.

	False detections							
Detector	10	31	50	65	78	95	167	422
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%	94.1%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2%	93.7%	–
Rowley-Baluja-Kanade	83.2%	86.0%	–	–	–	89.2%	90.1%	89.9%
Schneiderman-Kanade	–	–	–	94.4%	–	–	–	–
Roth-Yang-Ahuja	–	–	–	–	(94.8%)	–	–	–

Demo

https://medium.com/0xcode/the-viola-jones-face-detection-algorithm-3eb09055cfc2

Key Points

- 15x faster than any previous approach + more accurate
- Operates on multiple image scales
- Focuses on feature selection (allows for more complexity where needed)

A Convolutional Neural Network Cascade for Face Detection

2015

Overview

Current Issues:

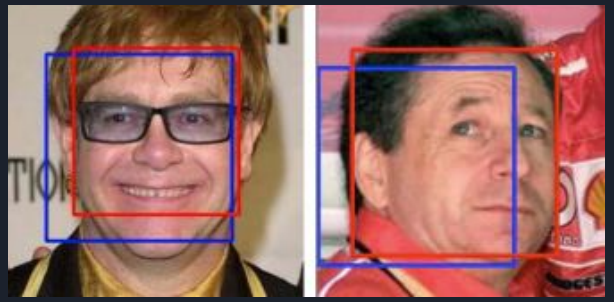
- Pose changes
- Exaggerated expressions
- Extreme illuminations

Lead to issues for face detector

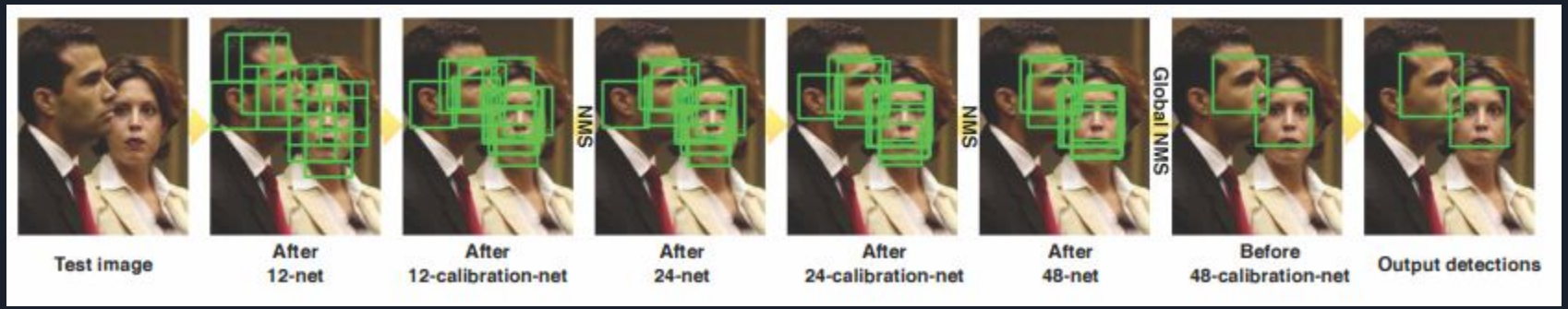
Solution:

- CNNs as it learns features to capture complex visual variations by using large training sets and can be generalized on GPU cores
- CNN cascades reduce computational expense because it rejects false detections quickly in early low resolution stages

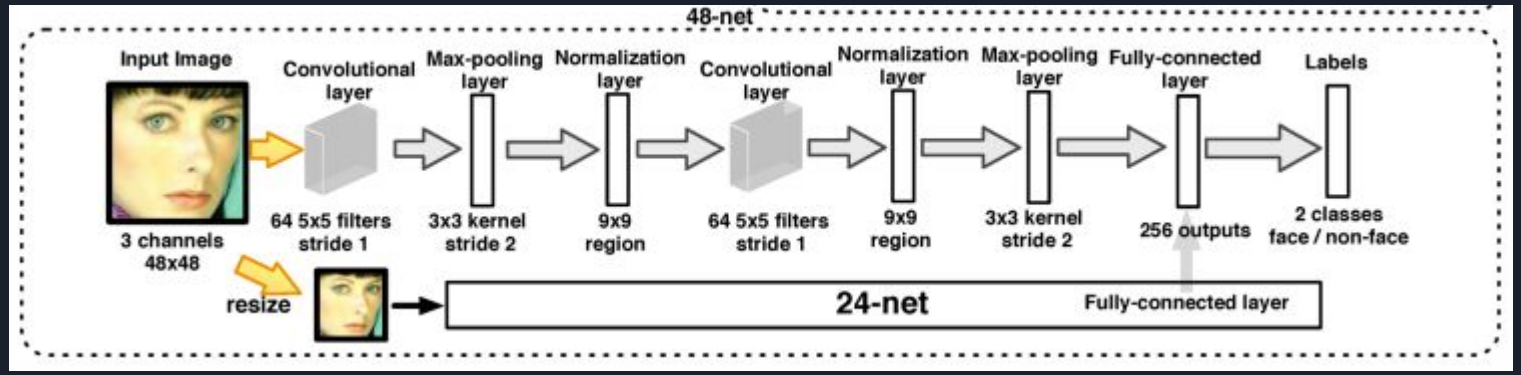
Architecture

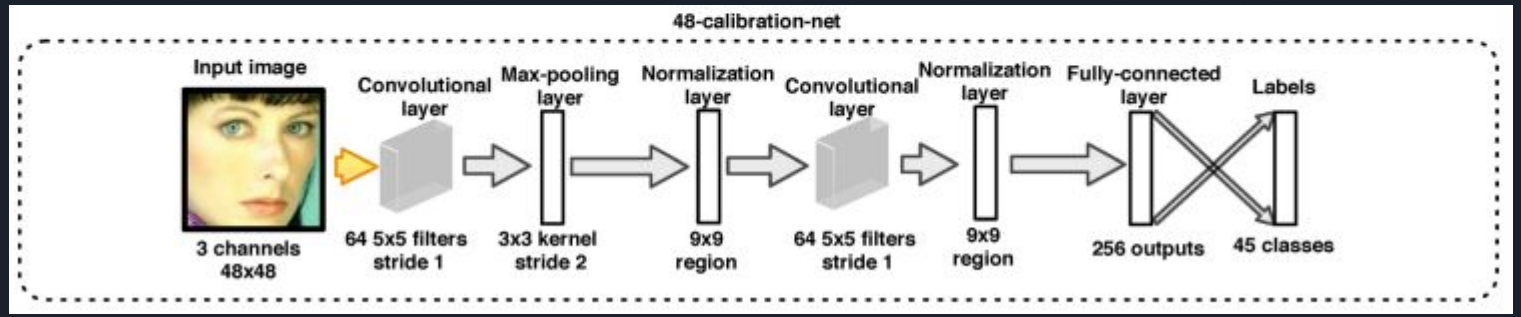


1. 12-net → scans whole image to quickly reject more than 90% of detection windows
2. 12-calibration-net → one-by-one as 12x12 images each ROI to adjust size and location for a potential nearby face
3. Non-maximum suppression (NMS) is applied to eliminate highly overlapped detection windows
4. 24-net → Resize into 24x24 images and repeat above process



Architecture (continued)





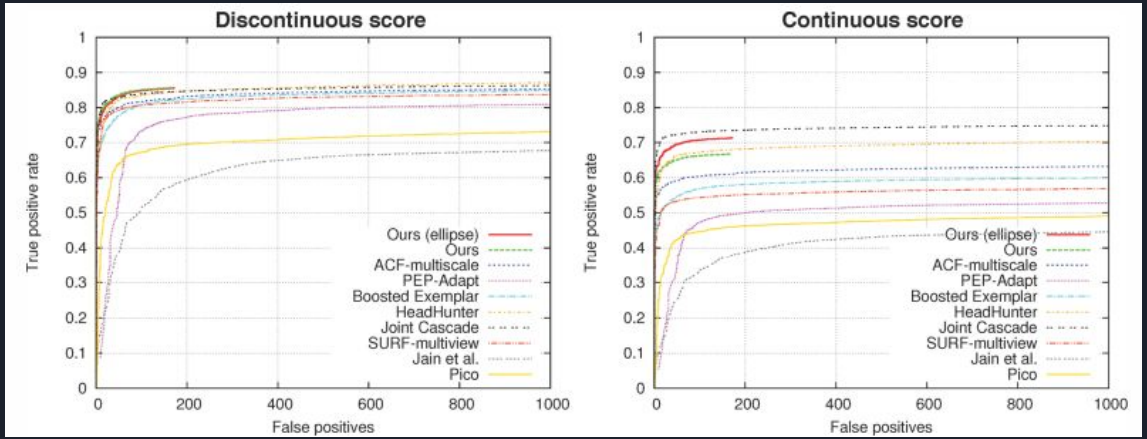
Results

Performance statistics of the cascade on FDDB (dataset)

Results

- 100 FPS on GPU
- Faster + Beats other state of the arc models

Stage	# windows	Recall
sliding window	5341.8	95.9%
12-*net*	426.9	93.9%
12-*calibration-net*	388.7	94.8%
24-*net*	60.5	88.8%
24-*calibration-net*	53.6	89.0%
48-*net*	33.3	85.8%
global NMS	3.6	82.1%
48-*calibration-net*	3.6	85.1%



Key Points

- Begins use of CNN cascade
- Quickly rejects non-face regions, so only computation will be on challenging regions at higher resolution
- Accelerates detection and improves bounding box quality

Joint Head Pose Estimation and Face Alignment Framework Using Global and Local CNN Features

2017

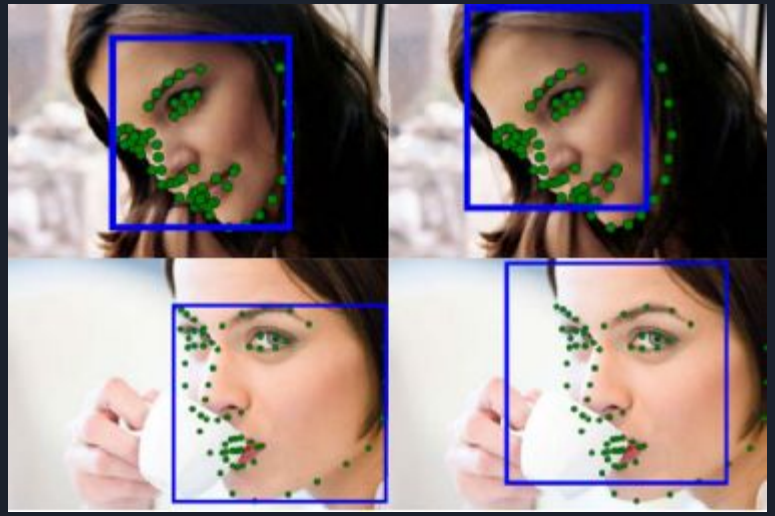
Overview

Bounding boxes are helpful for facial detection (left)

Face detectors make detection more difficult (right)

Contributions:

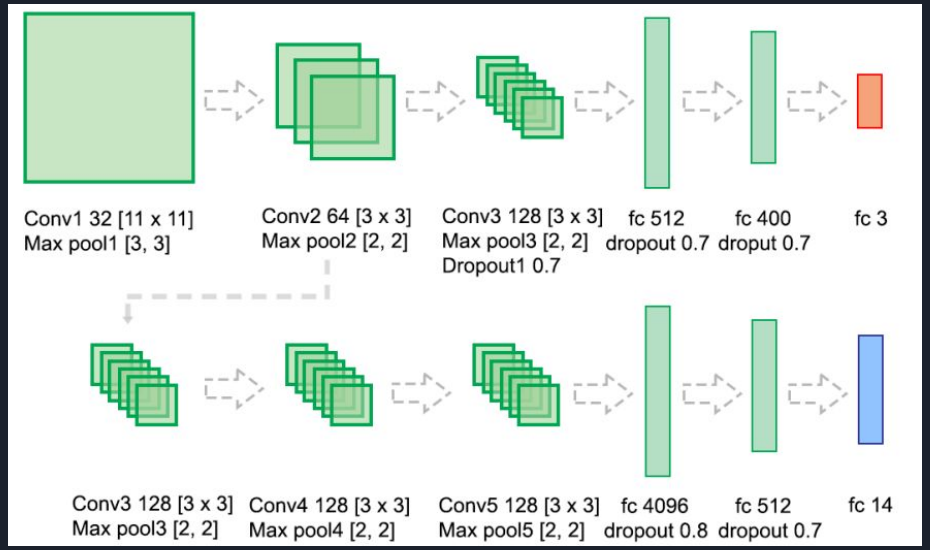
1. Leverages the relationship between head pose and landmarks for initialization
2. Explores the deep global and local features together via CNNs on the joint head pose estimation and face alignment in a cascaded way.



JFA System

Steps of the system:

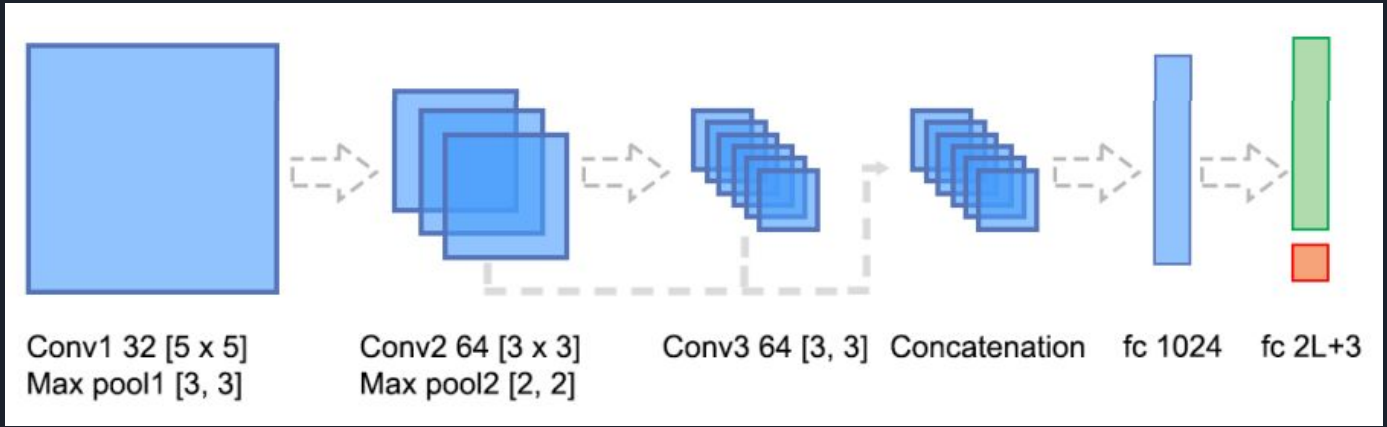
1. GNet Structure for initial head pose + primary landmark estimation



JFA System

Steps of the system:

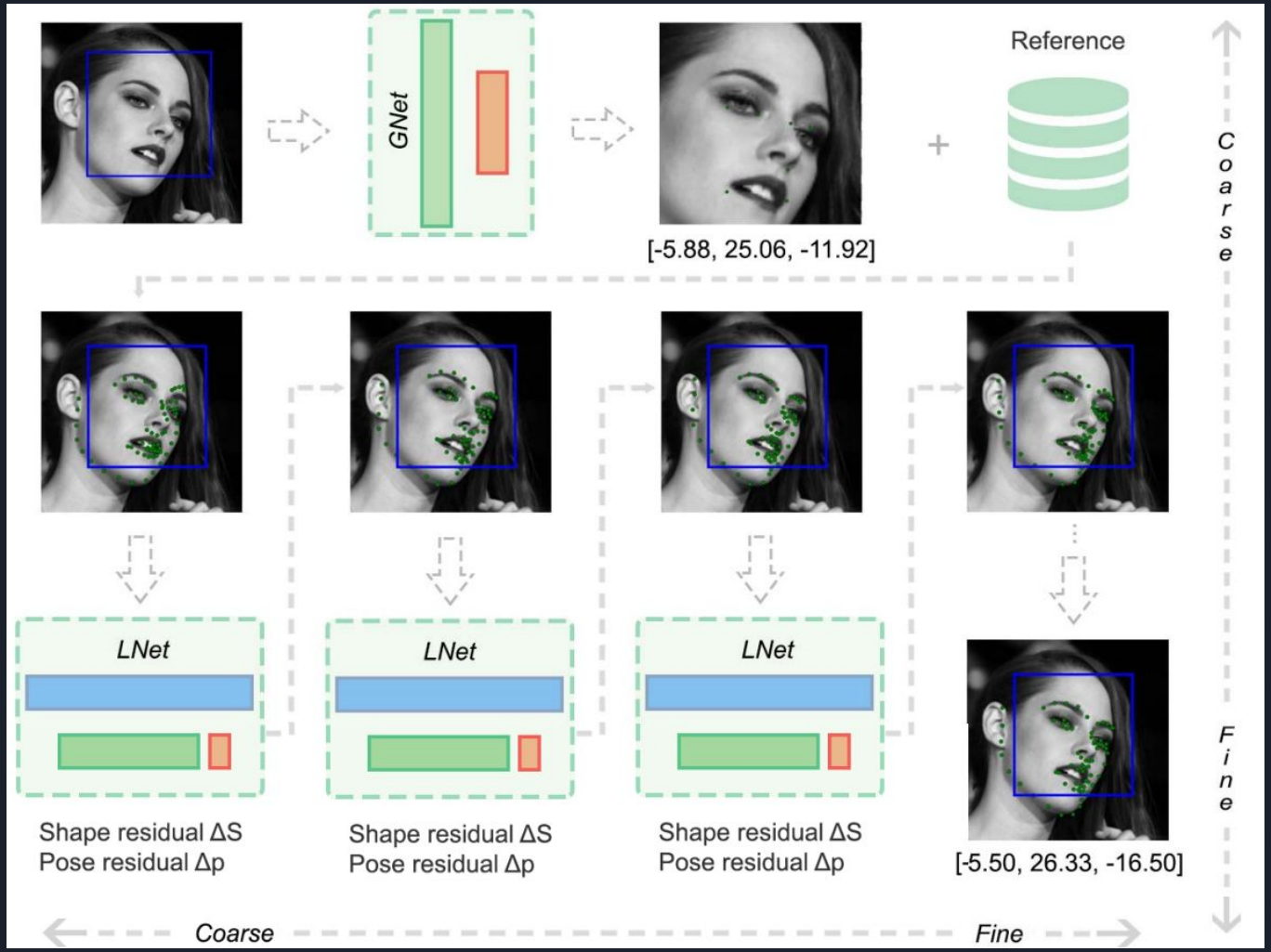
1. GNet Structure for initial head pose + primary landmark estimation
2. LNet structure for feature extraction



JFA System

Steps of the system:

1. GNet Structure for initial head pose + primary landmark estimation
2. LNet structure for feature extraction
3. Joint head pose estimation and face alignment learning (JFA algorithm)
    a. Based on coarse-to-fine principles exploring global and local feature
    b. From CNN features, it is used to update the shape for the next iteration


GNet
Reference
[-5.88, 25.06, -11.92]
LNet
LNet
LNet
Shape residual ΔS
Pose residual Δp
Shape residual ΔS
Pose residual Δp
Shape residual ΔS
Pose residual Δp
[-5.50, 26.33, -16.50]
Coarse
Fine
Coarse
Fine

Results

Rotation around

- Pitch - side-to-side axis
- Yaw - vertical axis
- Roll - front-to-back axis

Method	Pitch	Yaw	Roll
Yang *et al.* [28]	5.1	4.2	**2.4**
Random forest	4.7	5.5	4.8
SVR	4.8	7.8	5.3
GNet	3.5	3.3	2.6
JFA	**3.0**	**2.5**	2.6

With normalized mean root square error (NMRSE)

Method	Face detector	51 landmarks					68 landmarks				
		LFPW	HELEN	*Common*	*Challenge*	*Full*	LFPW	HELEN	*Common*	*Challenge*	*Full*
DRMF [2]	MATLAB	4.95	6.11	5.64	14.82	7.44	5.80	7.26	6.67	16.66	8.63
Chehra [3]	MATLAB	4.10	4.95	4.60	15.83	6.80	-	-	-	-	-
LBF [19]	OpenCV	4.63	5.69	5.26	18.58	7.87	5.58	6.58	6.18	18.94	8.68
ERT [13]	Dlib	**3.81**	**4.04**	**3.94**	12.17	**5.55**	**4.59**	**4.96**	**4.81**	13.66	6.55
3DDFA* [34]	Dlib	66.64	13.03	34.71	28.60	33.51	-	-	-	-	-
JFA	Dlib	4.65	5.26	5.01	**8.98**	5.79	5.08	5.48	5.32	**9.11**	**6.06**

Key Points

- Gathering only the face allows for easier detection
  - First rough estimate of face region/pose (**GNet**)
  - Learn local CNN features and predict shape/pose residuals (**LNet**)
  - Coarse-to-fine system refines shape/pose (**JFA**)
- Solves head pose estimation and landmark detection tasks jointly
- Outperforms other methods on head pose estimation task

img2pose: Face Alignment and Detection via 6DoF, Face Pose Estimation

2021

Overview

Landmark detectors are often optimized for bounding boxes by specific face detectors

Updating the face detector therefore requires re-optimizing the landmark detector

Having two successive components implies separately optimizing two steps of the pipeline for accuracy and – crucially for faces – fairness

6DoF

6DoF pose easier to estimate than detection landmarks (6 dimensions vs 10D or 136D)

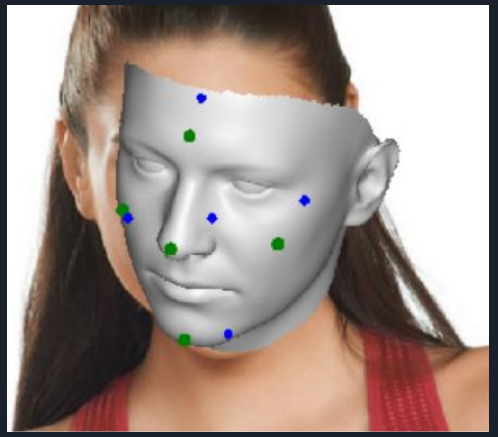
6DoF captures more than just bounding box locations

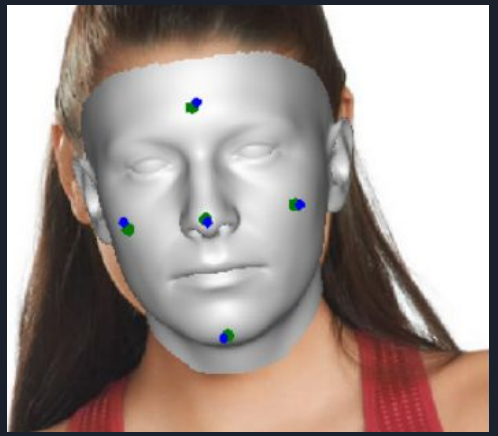
- Converts to a 3D-to-2D projection matrix
- Pose already captures the location of the face in the photo

3D face location via blue dots (ground truth)

Renders descending distances from the camera

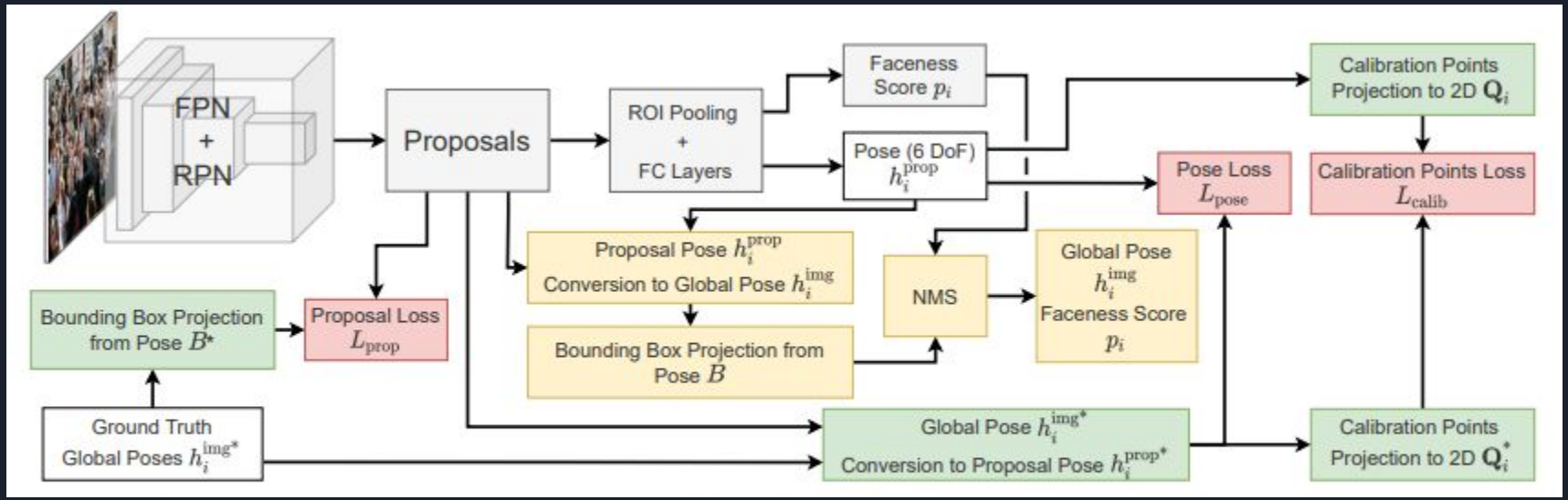
Does not use face bounding box labels





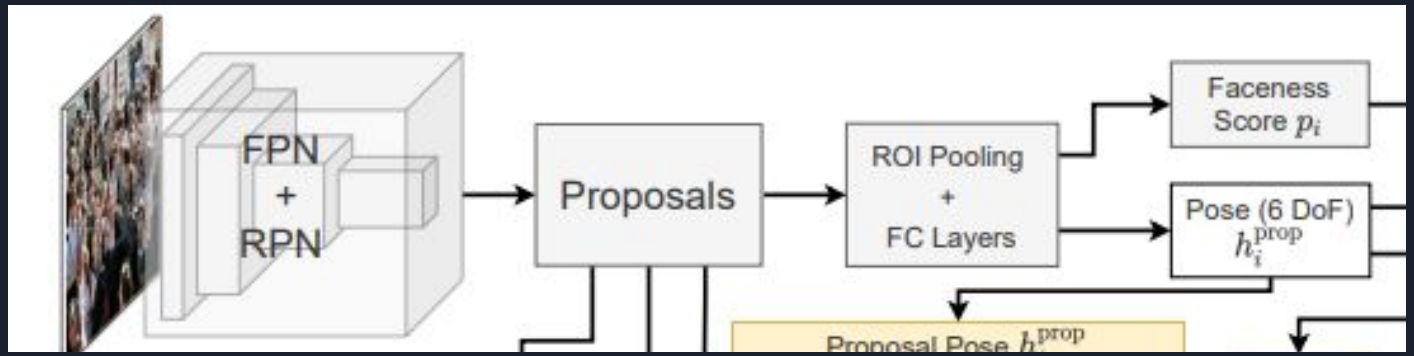
Architecture

Training Testing Default Components for Faster R-CNN



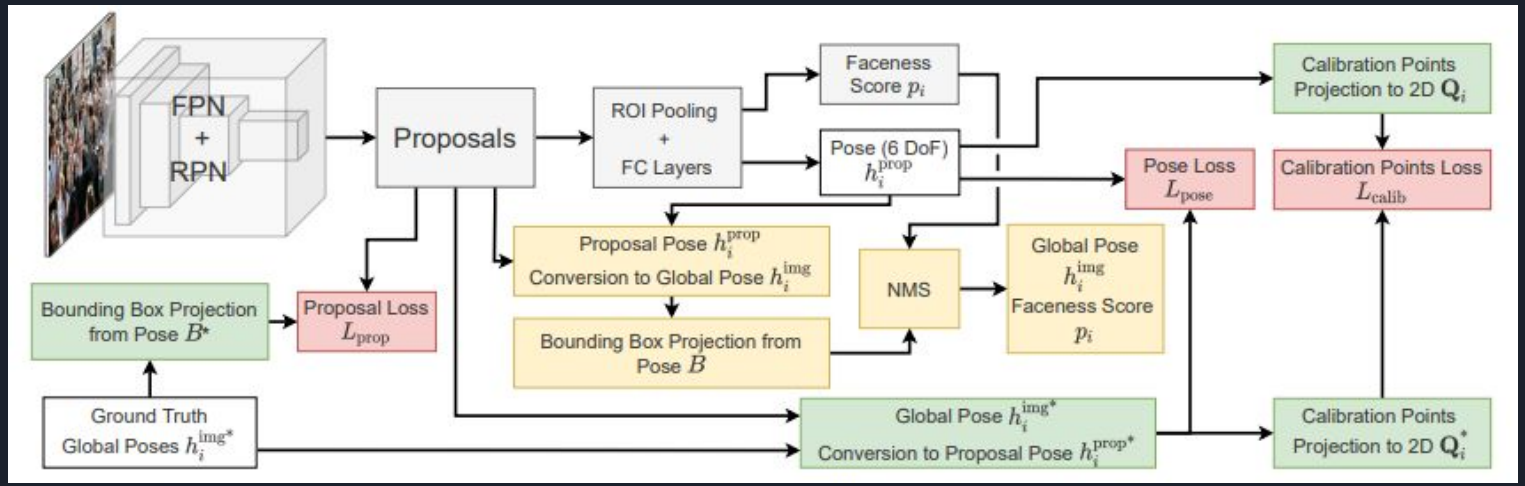
Architecture

1. Feature Pyramid using Faster R-CNN + region proposal network (RPN)
    a. Extracts features from each proposal with ROI pooling
    b. Passes into two different heads
        i. Standard face/no-face (facesness) classifier
        ii. Novel 6DoF face pose regressor



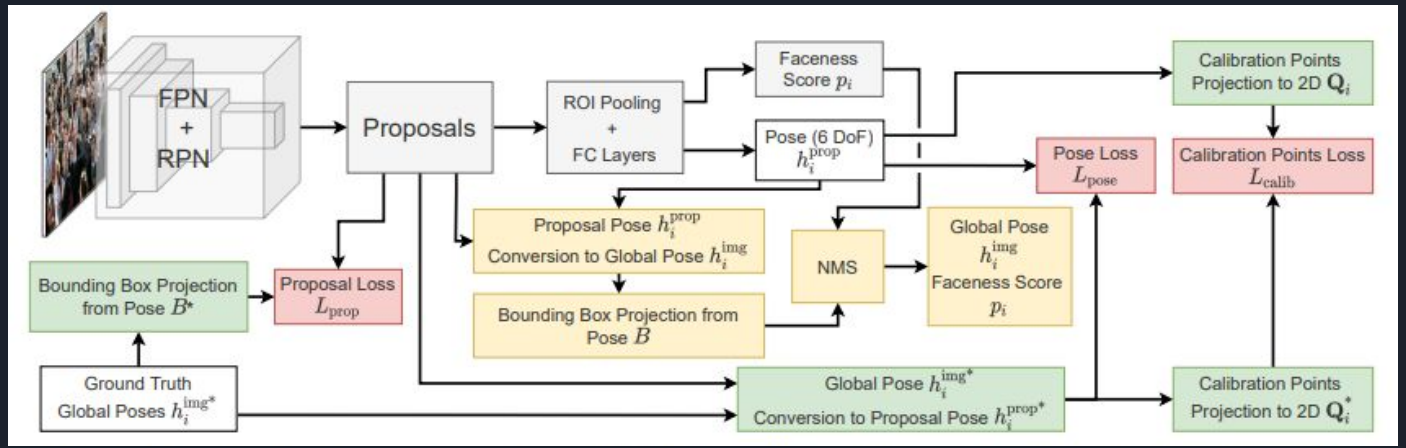
Architecture

1. Feature Pyramid using Faster R-CNN + region proposal network (RPN)
2. Pose Label Conversion
   a. Done from proposals not actual image
   b. Converts pose from local frames to global
   c. Converts pose from global to local frames



Architecture

1. Feature Pyramid using Faster R-CNN + region proposal network (RPN)
2. Pose Label Conversion
3. Training Losses
   a. Face classification loss
   b. Face pose loss
   c. Calibration point loss



Results

Against BIWI dataset →

(MAE → Mean Absolute Error)

Method	Direct?	Yaw	Pitch	Roll	MAE_r
Dlib (68 points) [33] †	✗	16.756	13.802	6.190	12.249
3DDFA [89] †	✗	36.175	12.252	8.776	19.068
FAN (12 points) [4] †	✗	8.532	7.483	7.631	7.882
Hopenet ($\alpha = 1$) [64] †	✗	4.810	6.606	3.269	4.895
QuatNet [31] †	✗	4.010	5.492	**2.936**	4.146
FSA-NET [81] †	✗	4.560	5.210	3.070	4.280
HPE [32] †	✗	4.570	5.180	3.120	4.290
TriNet [7] †	✗	**3.046**	4.758	4.112	3.972
RetinaFace R-50 (5 pnt.) [17]	✓	4.070	6.424	2.974	4.490
img2pose (ours)	✓	4.567	**3.546**	3.244	**3.786**

Against

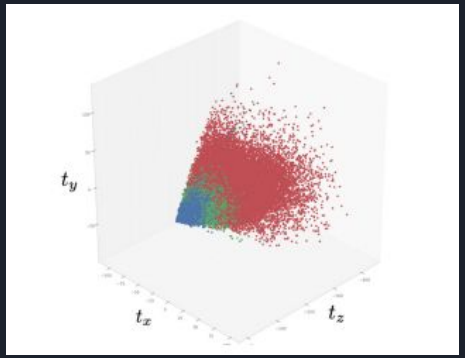
AFLW2000-3D →

Method	Direct?	Yaw	Pitch	Roll	MAE_r	X	Y	Z	MAE_t
Dlib (68 points) [33]	✗	18.273	12.604	8.998	13.292	0.122	0.088	1.130	0.446
3DDFA [89] †	✗	5.400	8.530	8.250	7.393	-	-	-	-
FAN (12 points) [4] †	✗	6.358	12.277	8.714	9.116	-	-	-	-
Hopenet ($\alpha = 2$) [64] †	✗	6.470	6.560	5.440	6.160	-	-	-	-
QuatNet [31] †	✗	3.973	5.615	3.920	4.503	-	-	-	-
FSA-Caps-Fusion [81]	✗	4.501	6.078	4.644	5.074	-	-	-	-
HPE [32] †	✗	4.870	6.180	4.800	5.280	-	-	-	-
TriNet [7] †	✗	4.198	5.767	4.042	4.669	-	-	-	-
RetinaFace R-50 (5 points) [17]	✓	5.101	9.642	3.924	6.222	0.038	0.049	0.255	0.114
img2pose (ours)	✓	**3.426**	**5.034**	**3.278**	**3.913**	**0.028**	**0.038**	**0.238**	**0.099**

Results (continued)

Wider Face Results

- Easy
- Medium
- Hard



Method	Backbone	Pose?	Validation Easy	Validation Med.	Validation Hard	Test Easy	Test Med.	Test Hard
SotA methods using heavy backbones (provided for completeness)								
SRN [11]	R-50	✗	0.964	0.953	0.902	0.959	0.949	0.897
DSFD [41]	R-50	✗	0.966	0.957	0.904	0.960	0.953	0.900
PyramidBox++ [68]	R-50	✗	0.965	0.959	0.912	0.956	0.952	0.909
RetinaFace [17]	R-152	✓*	0.971	0.962	0.920	0.965	0.958	0.914
ASFD-D6 [82]	-	✗	**0.972**	**0.965**	**0.925**	**0.967**	**0.962**	**0.921**
Fast / small backbone face detectors								
Faceboxes [85]	-	✗	0.879	0.857	0.771	0.881	0.853	0.774
FastFace [84]	-	✗	-	-	-	0.833	0.796	0.603
LFFD [29]	-	✗	**0.910**	0.881	0.780	0.896	0.865	0.770
RetinaFace-M [17]	MobileNet	✓*	0.907	0.882	0.738	-	-	-
ASFD-D0 [82]	-	✗	0.901	0.875	0.744	-	-	-
Luo *et al.* [46]	-	✗	-	-	-	**0.902**	0.878	0.528
img2pose (ours)	R-18	✓	0.908	**0.899**	**0.847**	0.900	**0.891**	**0.839**

Key Points

- Does not rely on first running a face detector or localizing facial landmark
- Maintains consistency of poses estimated for the same face across different image crops
- Faces have well-defined appearance statistics which can be relied upon for accurate pose estimation

Discussion Questions

Sources

- https://www.kairos.com/demos
- https://www.itpro.com/security/privacy/356882/the-pros-and-cons-of-facial-recognition-technology
- https://howthingsfly.si.edu/flight-dynamics/roll-pitch-and-yaw
- https://medium.com/0xcode/the-viola-jones-face-detection-algorithm-3eb09055cfc2