Introduction to Deep Learning for Facial and Gesture Understanding

Part IV: Facial Understanding

Raymond Ptucha,
Rochester Institute of Technology, USA

Tutorial
Nov 20, 2020, Noon-3pm

1

Fair Use Agreement

This agreement covers the use of all slides in this document, please read carefully.

- You may freely use these slides, if:
  - You send me an email telling me the conference/venue/company name in advance, and which slides you wish to use.
  - You receive a positive confirmation email back from me.
  - My name (R. Ptucha) appears on each slide you use.

2

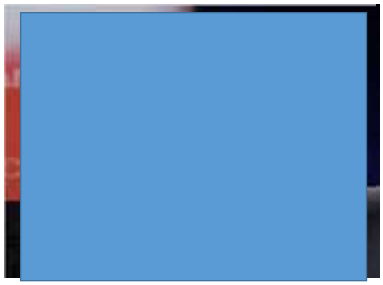
Agenda

- Part I: Introduction
- Part II: Convolutional Neural Nets
- Part III: Fully Convolutional Nets
- **Part IV: Facial Understanding**
- Part V: Hands-on code review

3

Facial Understanding

- Face Detection- find the faces in an image.
- Face Recognition- for all the faces found, identify who they are.
- Face Verification- determine if input image is that of a target person.
- Facial Feature Parts- location of nose, lips, eyes, etc.
- Facial Feature Points- outline of nose, lips, eyes, etc.

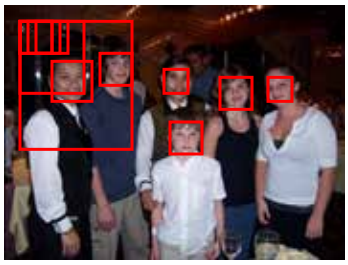


9

Holistic Face Detection

Using Viola-Jones, CVPR'01; Lienhart et al, DAGM'03

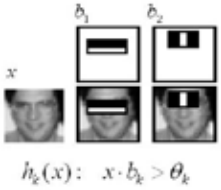
- The normalized image is scanned by sliding a (20×20 pixel) window across the image, computing a classifier H(X), determining if we have a face or not.



- With an initial image size of 384×256, we center our 20×20 window over 360×232 or 83,520 locations, computing H(X) at each location.
- We then increase our window (1.25×) looking for larger faces.
- For each match, the size of our window is the size of our face.

Viola Jones Face Detection Processing

Σ(white) – Σ(gray)



$h_k(x): \quad x \cdot b_k > \theta_k$

Resample to low resolution 384x288

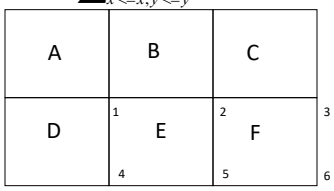
Integral Image (Papageorgiou)

Locate Haar-like basis functions

Cascade of progressively more complex features (Trained w/ AdaBoost)

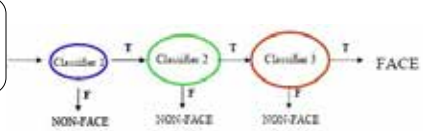
Filter multiple detections in same locale

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



Area(E) = ii(5) + ii(1) – ii(2) – ii(4)

A | B | C = ii(1) –[ii(2)-ii(1)] + [ii(3)-ii(2)]
= 2*ii(1) - 2*ii(2) + ii(3)



http://www.cs.utexas.edu/~grauman/courses/spring2007/395T/slides/Rapid_Object_Detection.pdf

CNN Face Detectors

- Deep learning techniques have replaced Viola Jones for face detection.
- Although more accurate, deep methods use orders of magnitude more memory and disk space.
- For example, Apple's CIDetector class from their Core Image framework was converted to deep methods in 2017.
  - When first introduced, used the fully convolutional OverFeat [Sermanet ICLR'14] method. Processing a 320×320 image with a 32×32 window, and stride of 16 would yield a 20×20 face map (CNN has ~20 layers).
  - Add a regression [x,y,w,h] output at each 20×20 location to better localize faces.
  - Use multi-stage image pyramid to process several resolutions of image with same framework.

Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks

Kaipeng Zhang[1] Zhanpeng Zhang[2] Zhifeng Li[1] Yu Qiao[1]

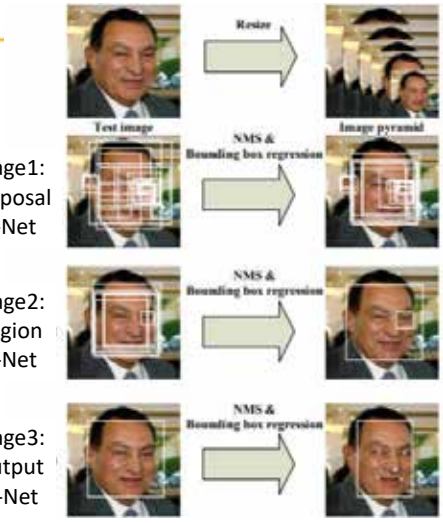
[1] Multimedia Research Center, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2] Multimedia Laboratory, Department of Information Engineering, The Chinese University of Hong Kong

IEEE Signal Processing Letters (SPL), vol. 23, no. 10, pp. 1499-1503, 2016

Abstract

Face detection and alignment in unconstrained environment are challenging due to various poses, illuminations and occlusions. Recent studies show that deep learning approaches can achieve impressive performance on these two tasks. In this paper, we propose a deep cascaded multi-task framework which exploits the inherent correlation between detection and alignment to boost up their performance. In particular, our framework leverages a cascaded architecture with three stages of carefully designed deep convolutional networks to predict face and landmark location in a coarse-to-fine manner. In addition, we propose a new online hard sample mining strategy that further improves the performance in practice. Our method achieves superior accuracy over the state-of-the-art techniques on the challenging FDDB and WIDER FACE benchmarks for face detection, and AFLW benchmark for face alignment, while keeps real time performance.

https://kpzhang93.github.io/MTCNN_face_detection_alignment/



Stage1: Proposal P-Net

Formulate pyramid of resolutions

Fully convolutional network which finds facial bounding boxes: [P(face),UL,UR,width,height]
Use non-maximal suppression (NMS) to clean up boxes

Stage2: Region R-Net

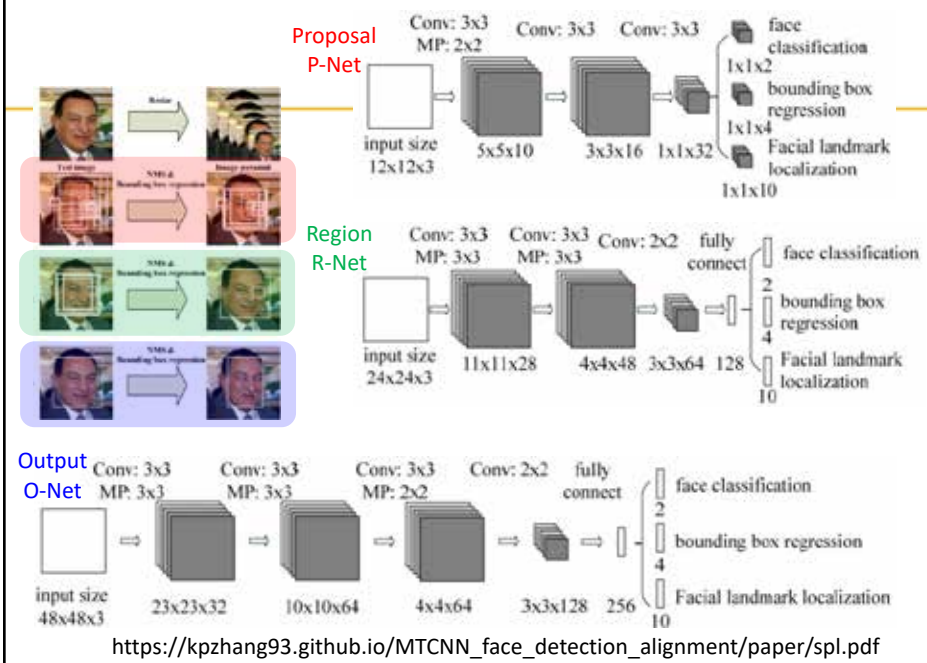
Candidate proposals further scrutinized, rejecting false positives and more accurately locating true positives. Another NMS stage.

Stage3: Output O-Net

Final check of proposal areas. In addition to facial bounding box, also return centers of eyes, nose tip, and corners of mouth.

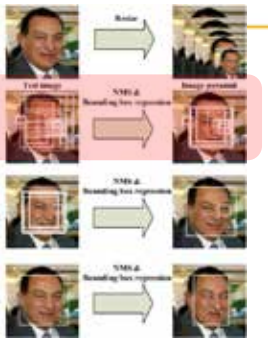
https://kpzhang93.github.io/MTCNN_face_detection_alignment/paper/spl.pdf

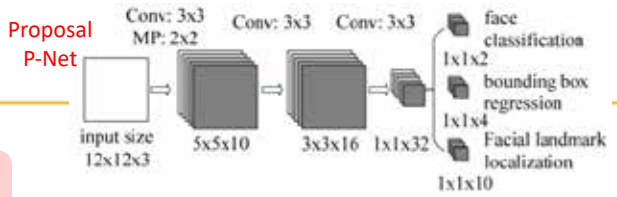
18



https://kpzhang93.github.io/MTCNN_face_detection_alignment/paper/spl.pdf

19



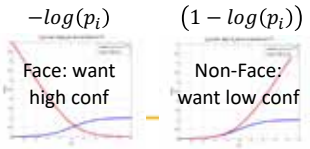


What about the pyramid?

- A 12x12 region would give one output, face vs no face
- A 20x20 image would give 5x5 face predictions (20→18→9→7→5)
  - Each of these 25 locations returns {p(face), x,y,w,h,5 points}
- A 224 image would give 224→222→111→109→107 or approx. 10K locations

https://kpzhang93.github.io/MTCNN_face_detection_alignment/paper/spl.pdf

Learning Objectives



- Face classification done as two-class objective function:

$$L_i^{det} = -(y_i^{det}\log(p_i) + (1-y_i^{det})(1-\log(p_i)))$$

y_i^{det} is GT location
p_i is predicted face from nnet

- Bounding box: Predict left, top, height, width. Regression problem with Euclidean loss:

$$L_i^{box} = \left\|\hat{y}_i^{box} - y_i^{box}\right\|_2^2$$

y_i^{box} = GT locations
$\hat{y}_i^{box}$ = estimate locations

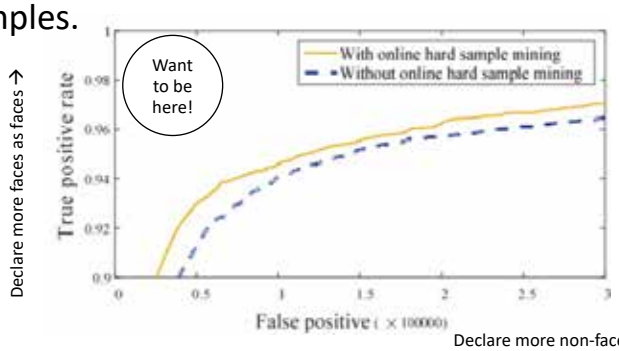
- Facial landmark: Predict left/right eye, nose tip, left/right mouth. Regression problem with Euclidean loss:

$$L_i^{landmark} = \left\|\hat{y}_i^{landmark} - y_i^{landmark}\right\|_2^2$$

$y_i^{landmark}$ = GT
$\hat{y}_i^{landmark}$ = estimate

Online Hard Sample Mining

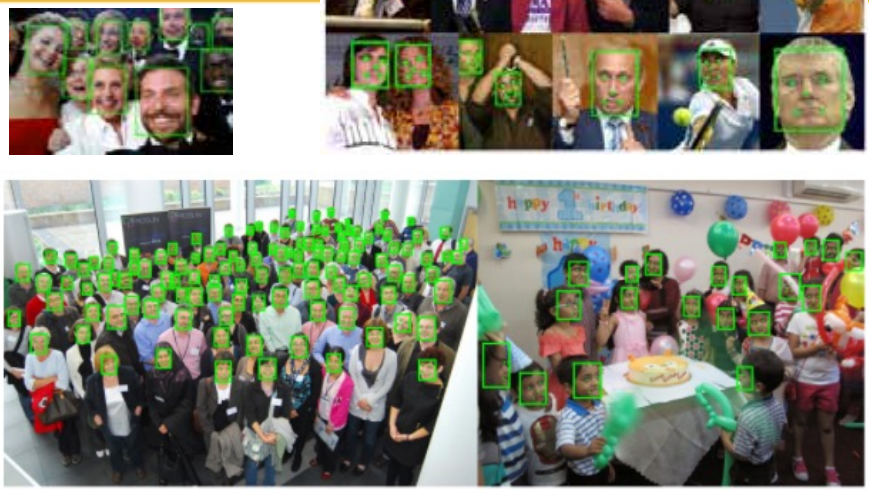
- For each minibatch, the hard samples are defined as the top 70% samples with the highest losses.
- Only compute gradient descent using the hard samples.



R. Ptucha '20 22

22

MTCNN Results



https://kpzhang93.github.io/MTCNN_face_detection_alignment/

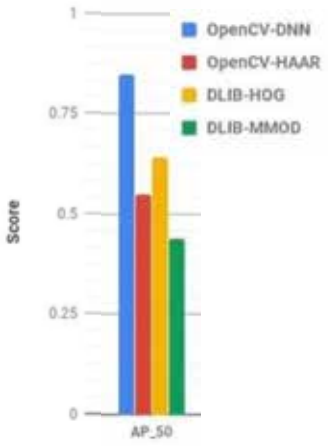
R. Ptucha '20 23

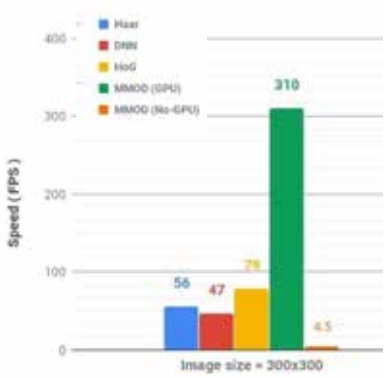
23

Face Detection: OpenCV vs. Dlib

https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/

FDDB Dataset



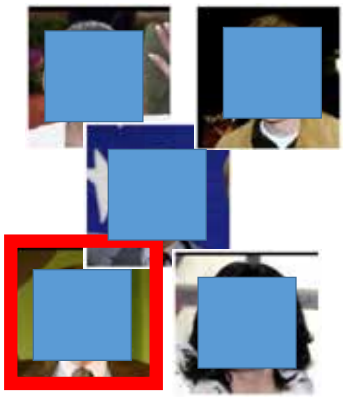


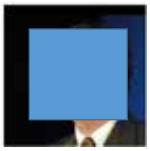
R. Ptucha '20 24

24

One-shot Learning for Face Recognition

- Given a single exemplar of a face, can you recognize future exemplars.





Bill Gates!

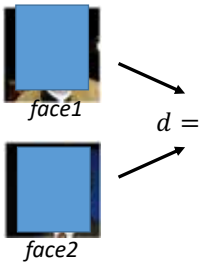
Images from LFW, Kawulok et al. 2016

R. Ptucha '20 25

25

Face Recognition: Similarity Function

- Can we learn a function *d(f(face1),f(face2))* which returns a small value when faces are from same person and large value when faces from different people?



$$d = \|f(x^{(1)}) - f(x^{(2)})\|_2^2$$

Perhaps the simplest method is to compare two images, pixel by pixel, comparing to a threshold.

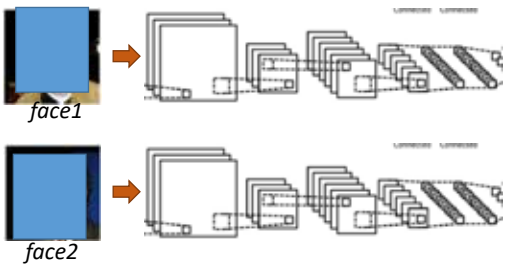
Unfortunately, this method is not robust to lighting, pose, background, etc.

Similarity Function

FaceNet: Schroff et al. 2015

- Can we learn a function *d(f(face1),f(face2))* which returns a small value when faces are from same person and large value when faces from different people?

Siamese network- identical networks



Instead, use a CNN encoding (face2vec)

$$d(f(face1), f(face2))$$

$$d = \|f(x^{(1)}) - f(x^{(2)})\|_2^2$$

Similarity Function Objective

Anchor (A) Positive (P) Anchor (A) Negative (N)

$d\big(f(A), f(P)\big)$ $d\big(f(A), f(N)\big)$

Triplet loss (A,P,N)

Want: $\|f(A) - f(P)\|_2^2 + \alpha \leq \|f(A) - f(N)\|_2^2$ Add margin

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha \leq 0$$

29

Similarity Function Objective

Given three images, Anchor (A), Positive (P), Negative (N):

Want: $\|f(A) - f(P)\|_2^2 + \alpha \leq \|f(A) - f(N)\|_2^2$

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha \leq 0$$

$$\mathcal{L}(A, P, N) = max(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha, 0)$$

Want loss to be as small as possible, so as long as this part is negative, the loss is 0.

30

Similarity Function Objective

Given three images, Anchor (A), Positive (P), Negative (N):

Want: $\|f(A) - f(P)\|_2^2 + \alpha \leq \|f(A) - f(N)\|_2^2$

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha \leq 0$$

$$\mathcal{L}(A, P, N) = max(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha, 0)$$

Overall cost: $\mathcal{J} = \sum_{i=1}^{n} \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$

n is images in minibatch

R. Ptucha '20 31

31

How to Choose A, P, N

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha \leq 0$$

- If A, P, N are chosen randomly, $d(A, P) + \alpha \leq d(A, N)$ then is too easily satisfied.
- Need to find "hard" triplets, given A:
  - Hard positive: $\underset{P}{\text{argmax}} \|f(A) - f(P)\|_2^2$
  - Hard negative: $\underset{N}{\text{argmin}} \|f(A) - f(N)\|_2^2$

A: Anchor
P: Positive
N: Negative

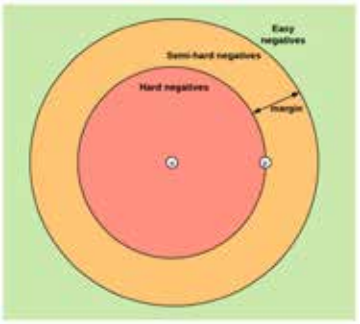
- Infeasible to find overall argmin/max, so choose over mini-batches in an online fashion.
  - A few thousand faces per mini-batch.
  - Data sampled such that ~40 faces/identity in each mini-batch.
  - Facenet uses all (A,P) pairs, but only uses hardest (A,N) pairs, but removes α to avoid bad local minima early in training.

R. Ptucha '20 32

32

Hard Negative Mining

- Negatives should be at least margin away positive samples
- FaceNet uses all (A,P) pairs, but only uses hardest (A,N) pairs…



https://omoindrot.github.io/triplet-loss

- Batch all: Average the loss on the hard and semi-hard triplets.
  - Do not take into account the easy triplets, as averaging on them would make the overall loss very small
- Batch hard: For each anchor, select the hardest positive (biggest distance d(a,p)) and the hardest negative (smallest distance d(a,n)) among the batch.
- In general, batch hard better.
- Lots of variants which, for example exclude outliers.

How to Choose A, P, N

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha \leq 0$$

- If A, P, N are chosen randomly, $d(A,P) + \alpha \leq d(A,N)$ then is too easily satisfied.
- Need to find "hard" triplets, given A:
  - Hard positive: $\underset{P}{\operatorname{argmax}} \|f(A) - f(P)\|_2^2$
  - Hard negative: $\underset{N}{\operatorname{argmin}} \|f(A) - f(N)\|_2^2$

A: Anchor
P: Positive
N: Negative

- This will force $d(A,P)$ to go small and $d(A,N)$ to go large.
- Through backpropagation, the network will continually adjust its weights to minimize our cost.
- FaceNet used 100M-200M training face thumbnails of about 8M faces.

Contrastive Loss

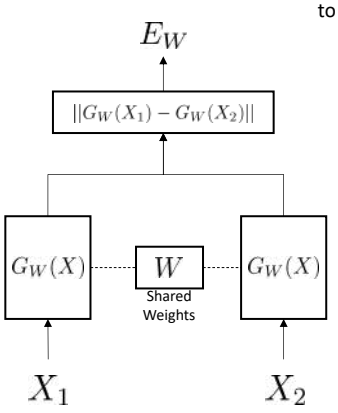
Want E_W to be low

Similar pairs y==1

Dissimilar pairs y==0

$$E_W = \frac{1}{2N}\sum_{n=1}^{N}(y)d^2 + (1-y)max(margin-d,0)^2)$$

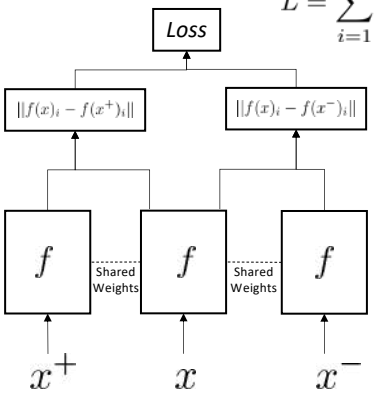
$$d = ||G_W(X_1) - G_W(X_2)||$$



- N is the no. of sample pairs
- y is a binary indicator. 1 for same and 0 for dissimilar pairs
- W are learned weights in a network G
- *margin* is a hyper-parameter
- y==1: Gradient descent will force X_1, X_2 same
- y==0: Gradient descent will force: margin-d < 0; or d > margin

Triplet Loss

$$L = \sum_{i=1}^{N}\left[||f(x)_i - f(x^+)_i||^2 - ||f(x)_i - f(x^-)_i||^2 + \alpha\right]_+$$



- This loss makes sure that, given an anchor point, the projection of a positive point is closer to the anchor's projection than that of a negative point belonging to another class, by at least a margin
- N is the total number of triplets (x, x^+, x^-) in the dataset
- α is the margin hyper-parameter
- Selecting hard negatives is challenging hence semi-hard negatives are chosen (more in the reference)
- $[.]_+$ denotes the hinge function which takes the positive component of the argument. It ensures loss only contributes when positive.

Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Lifted Structured Loss

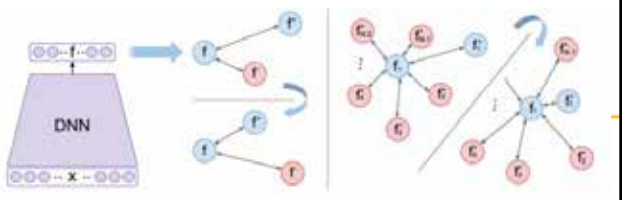
- Each positive pair compares the distances against all the negative pairs weighted by a margin constraint.
- The idea is to have a differentiable smooth loss which incorporates the online hard negative mining functionality using the log-sum-exp formulation.

$$L_{i,j} = \frac{1}{2|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \Big[log\Big(\sum_{(i,k)\in\mathcal{N}} exp\{\alpha - D_{i,k}\} + \sum_{(j,l)\in\mathcal{N}} exp\{\alpha - D_{j,l}\} \Big) + D_{i,j} \Big]_+^2$$

$\mathcal{P}$ denotes the set of pairs of examples with same labels
$\mathcal{N}$ denotes the set of pairs of examples with different class labels

Song, Hyun Oh, et al. "Deep metric learning via lifted structured feature embedding." *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016.

N-Tuple Loss



Triplet Loss N-Tuple Loss

- It generalizes triplet loss by allowing joint comparison among more than one negative examples.
- Also extended to multi-class N-pair loss.

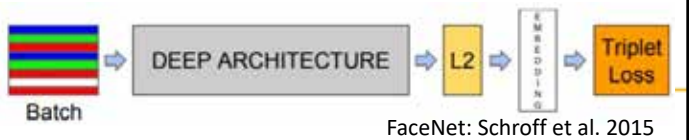
$$L(\{x, x^+, \{x_i^-\}_{i=1}^{N-1}\}; f) = log\Big(1 + \sum_{i=1}^{N-1} exp(f^\top f_i^- - f^\top f^+)\Big)$$

$\{x, x^+, x_1^-, x_2^-, ..., x_{N-1}^-\}$ is a (N+1) tuple for training

f is a embedding kernel defined by a deep neural network

Sohn, Kihyuk. "Improved deep metric learning with multi-class n-pair loss objective." *Advances in Neural Information Processing Systems*. 2016.

FaceNet



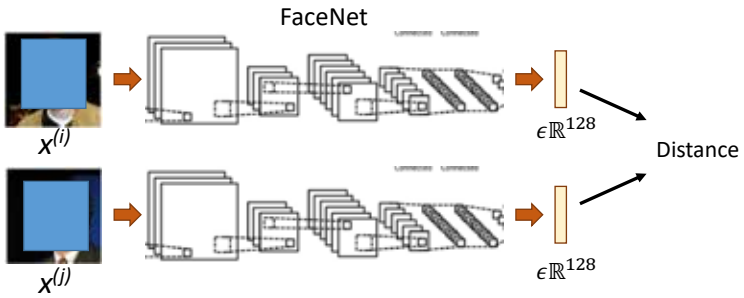
FaceNet: Schroff et al. 2015

- Learns a 128 byte embedding per face.
- The learned embedding is useful for face verification, facial recognition, and clustering of similar faces.
- Embedding faces to this latent representation gives 99.63% on Labeled Faces in the Wild and 95.12% on YouTube Faces database.
- Rather than construct a classification CNN, FaceNet uses triplet loss with online negative exemplar mining.
- Faces are tightly cropped faces using [Chen et al. ECCV'14] , no 2D or 3D alignment.

39

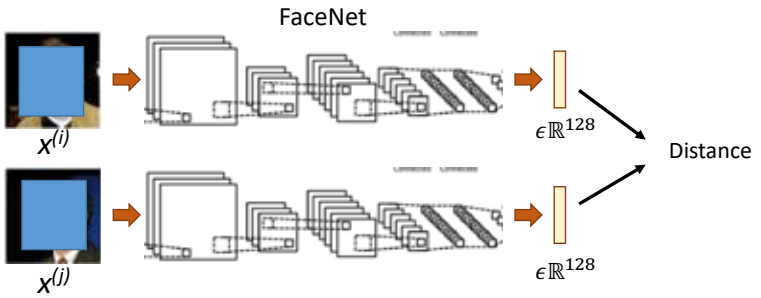
Using FaceNet

- Given a face, FaceNet outputs a 128 dimensional vector, a fingerprint of a face.
- If distance between two vectors close, we have a match.



40

Using FaceNet for Face Verification

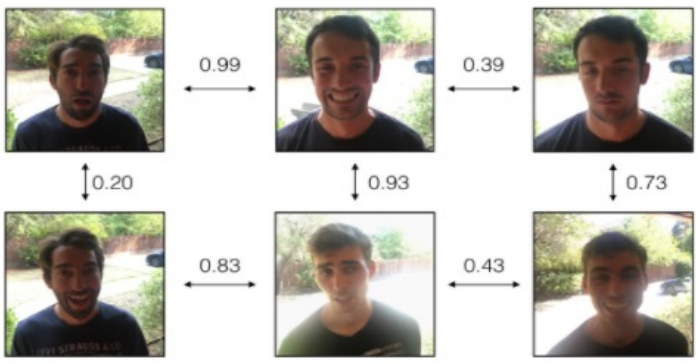


Elementwise comparison of each dimension:

$$Distance = \sum_{i=1}^{128} \left\| f\left(x^{(i)}\right)_k - f\left(x^{(j)}\right)_k \right\|^2$$

Example of FaceNet for Face Verification

Example distances outputs between three individuals' FaceNet encodings



Deeplearning.ai W4C4 programming assignment

Embedding Dimensionality

#dims	VAL
64	86.8% ± 1.7
128	87.9% ± 1.9
256	87.7% ± 1.9
512	85.6% ± 2.0

- 128 dimensional float actually used, but experiments show can be quantized to 128 bytes.
- Larger dimensions probably need more training data to justify.

Training Set Size

#training images	VAL
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

FaceNet: Schroff et al. 2015

- That is a lot of training data!

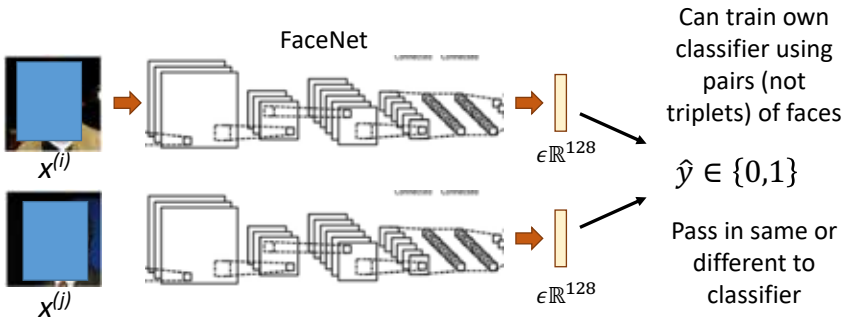
VGGFace2- A Large Scale Facial Recognition Dataset

- 3.3 million faces
- 9,000 identities
- 87 to 843 samples per subject (avg = 362)

http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/

Using FaceNet on your own Dataset

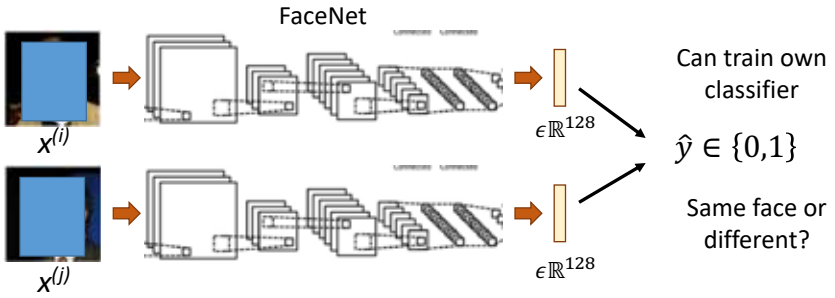
- Given a face, FaceNet outputs a 128 dimensional vector, a fingerprint of a face.
- You can use FaceNet on your own dataset:



R. Ptucha '20 46

46

Using FaceNet for Face Verification



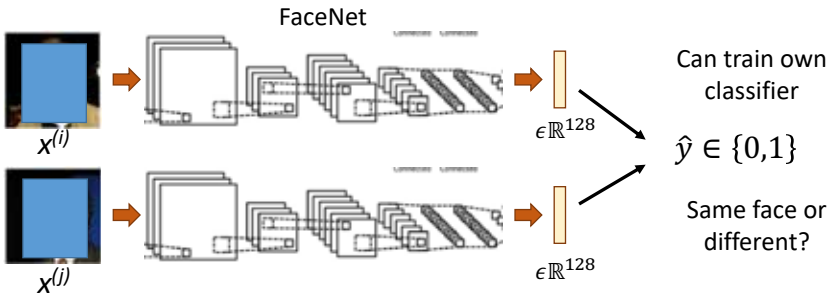
Elementwise comparison of each dimension:

$$\hat{y} = \sigma\left(\sum_{i=1}^{128} \left| f\left(x^{(i)}\right)_k - f\left(x^{(j)}\right)_k \right|\right)$$

R. Ptucha '20 47

47

Using FaceNet for Face Verification

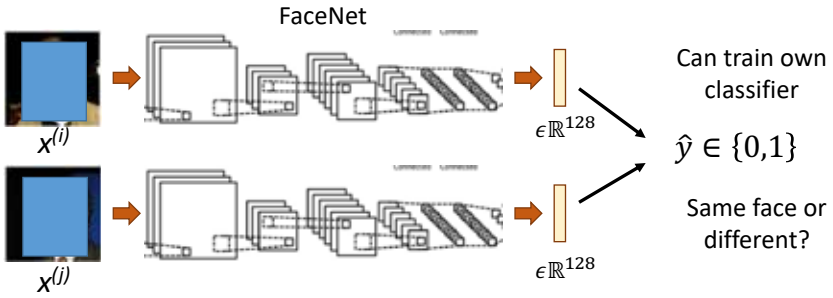


Can fit, for example into a logistic regression:

$$\hat{y} = \sigma\left(\sum_{i=1}^{128} w_i \left| f\left(x^{(i)}\right)_k - f\left(x^{(j)}\right)_k \right| + b\right)$$

48

Using FaceNet for Face Verification

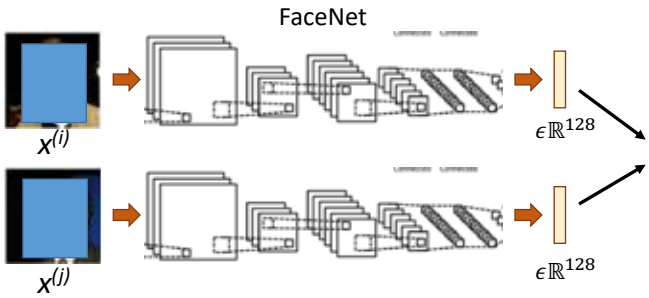


Sometimes use Chi-square fit (normalize by distance):

$$\hat{y} = \sigma\left(\sum_{i=1}^{128} w_i \left[\frac{\left(\left(x^{(i)}\right)_k - \left(x^{(j)}\right)_k\right)^2}{\left(x^{(i)}\right)_k + \left(x^{(j)}\right)_k}\right] + b\right)$$

49

Using FaceNet for Face Verification



- At runtime, expensive to run both $x^{(i)}$ and $x^{(j)}$ through FaceNet for each sample in dataset.
- Typically precompute 128 vector value for all training faces to speed up runtime.

Facial Feature Points

- Constrained Local Models CLMs such as active shape/appearance models model the appearance of each facial landmark individually, then use a shape model to constrain location, then iteratively repeat.
- CLMs have difficulty with variations in expression, illumination, facial hair, makeup, and accessories.
- Given enough data, nnets can model these complex variations.

Zadeh et al. CVPR'17

Traditional Approach

Viola Jones Face Detection

Search for actual point locations using Mahalanobis distance

Repeat ~3-5×

Average eye and 82 facial feature points

Restrict based on PCA statistics

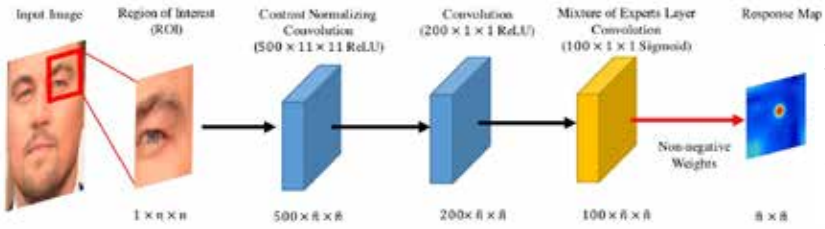
R. Ptucha '20

52

52

Convolutional Experts Constrained Local Model (CE-CLM)

Zadeh et al. CVPR'17



This is followed by a point distribution model which constrains landmark locations.

Start with face bbox. Extract *n*×*n* ROI around best guess of where landmark is. (*n*=19). Best guess determined by model.

Learn 500 11×11×1 filters. No padding used, so width and height reduced by 10.

Learn 200 1×1×500 filters.

Learn 100 1×1×200 filters. Each neuron and expert- uses sigmoid, which estimates 100 to 0:1 "probabilities".

Learn 100 non-negative weights at each pixel position to form final response map.

R. Ptucha '20

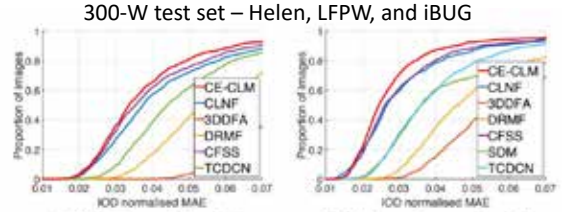
53

53

CE-CLM: State-of-the-art Results

All training done with Multi-PIE and 300W

300-W test set – Helen, LFPW, and iBUG



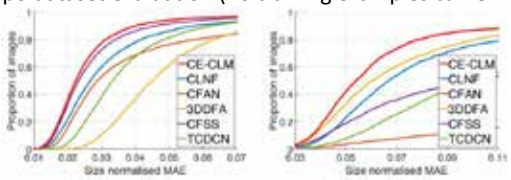
(a) With face outline (68) (b) Without face outline (49)

68 points

(only use eyebrows, eyes, lips, nose)

Mean Average Error (MAE) results normalized by Inter-Ocular-Distance (IOD)

Menpo dataset evaluation (no training examples came from Menpo)

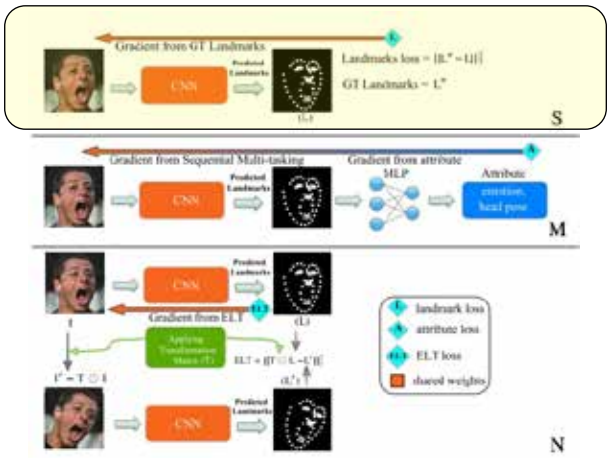


(a) Frontal faces (b) Profile faces

Zadeh et al. CVPR'17

Improving Landmark Localization with Semi-Supervised Learning

Honari et al. CVPR'18

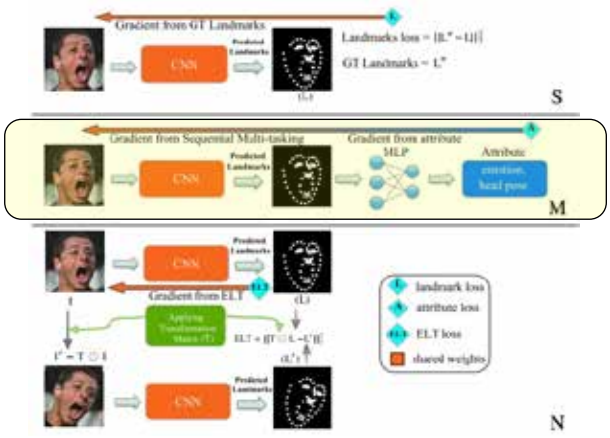


- Standard landmark labeling using ground truth labelled images.
- Typical CNN with (68) X-Y landmarks.
- Solve regression-type cost function for each point.

http://openaccess.thecvf.com/content_cvpr_2018/html/Honari_Improving_Landmark_Localization_CVPR_2018_paper.html

Improving Landmark Localization with Semi-Supervised Learning

Honari et al. CVPR'18



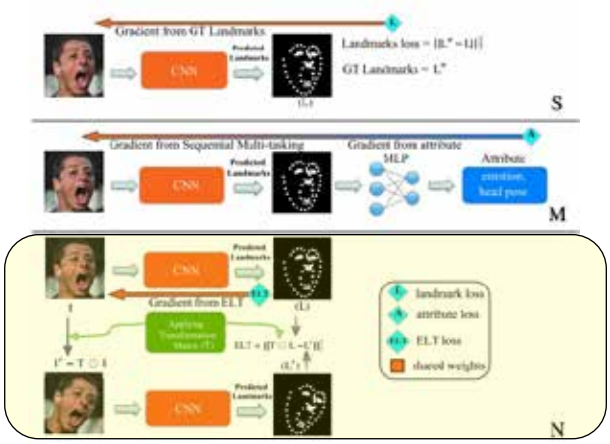
- Feed predicted landmarks into Attribute MLP.
- Error from Attribute is backpropagated through entire CNN.

http://openaccess.thecvf.com/content_cvpr_2018/html/Honari_Improving_Landmark_Localization_CVPR_2018_paper.html

56

Improving Landmark Localization with Semi-Supervised Learning

Honari et al. CVPR'18



- Pass image through CNN, getting predicted points, L.
- Apply transform to both image (I') and L landmarks (T⊙L).
- Pass I' through CNN to get L'.
- Backprop difference between T⊙L and L'.
- Note: S << M ≤ N

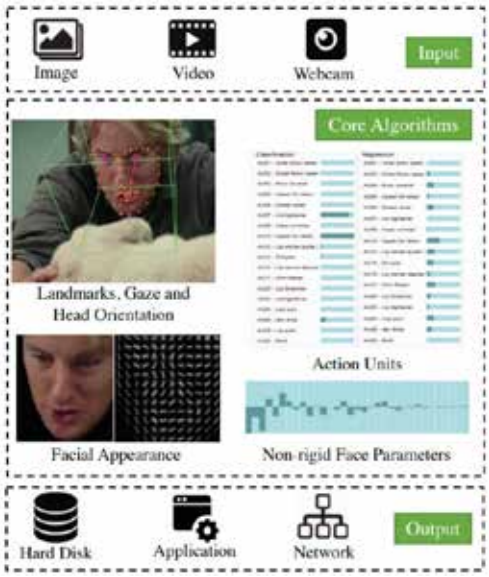
http://openaccess.thecvf.com/content_cvpr_2018/html/Honari_Improving_Landmark_Localization_CVPR_2018_paper.html

57

OpenFace 2.0

https://github.com/TadasBaltrusaitis/OpenFace

- Free, open source software with both pretrained binaries and source code.
- Facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation.
- Real-time performance and state-of-theart results in all categories.



OpenFace 2.0, Baltrusaitus et al. F&G'18

58

OpenFace 2.0

OpenFace 2.0, Baltrusaitus et al. F&G'18
https://github.com/TadasBaltrusaitis/OpenFace

- Face detection:
  - Multi-Task Cascaded CNN (MTCNN), Zhang et al. '16
- Facial landmark detection:
  - Convolutional Experts Constrained Local Model (CE-CLM), Zadeh et al. '17
- Head pose:
  - CE-CLM represents facial landmarks in 3D and projects them to the image. Given landmarks, solve *n* point perspective problem.
- Eye gaze:
  - Use a Constrained Local Neural Field (CLNF) landmark detector [Wood et al. 2015] to detect eyelids, iris, and the pupil. Solve pupil center on eye-ball sphere in 3D. Gaze is vector from the 3D eyeball center to the 3D pupil center.

59

OpenFace 2.0

OpenFace 2.0, Baltrusaitus et al. F&G'18
https://github.com/TadasBaltrusaitis/OpenFace

- Facial Action Units (AU):
  - Implementation of [Baltrusaitis et al. FG'15] which uses HOG and CE-CLM features, passed into SVM model. (seems to work just as well as deep models…)
  - Intensity and presence predicted for all 18 AUs, except for AU28, for which only presence predicted.

AU	Full name	Illustration
AU1	Inner brow raiser	
AU2	Outer brow raiser	
AU4	Brow lowerer	
AU5	Upper lid raiser	
AU6	Cheek raiser	
AU7	Lid tightener	
AU9	Nose wrinkler	
AU10	Upper lip raiser	
AU12	Lip corner puller	
AU14	Dimpler	
AU15	Lip corner depressor	
AU17	Chin raiser	
AU20	Lip stretched	
AU23	Lip tightener	
AU25	Lips part	
AU26	Jaw drop	
AU28	Lip suck	
AU45	Blink	

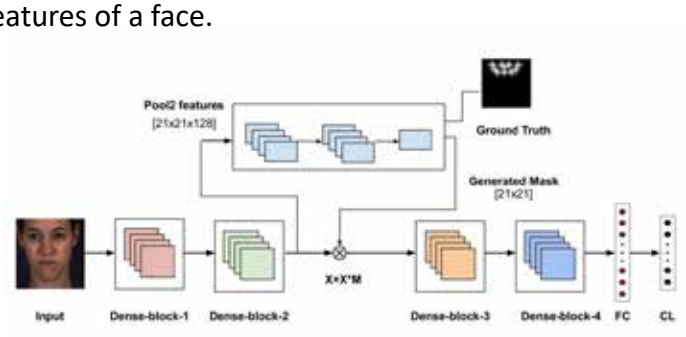
Megvii and Face++

https://www.faceplusplus.com/

- Megvii is arguably the world leader in facial recognition technology.
- Face++ uses 106 facial feature points.
- Open developer platform with 300K developers in 150 countries (pay to use toolkit)
- Also does body understanding/tracking

Learning Guided Attention Masks for Facial Action Unit Recognition, Lakshminarayana et al. FG2020

- Multi-task learning: Jointly learn attention and AU classification.
- Using human visual fixations, a guided attention mechanism facilitates the network to 'look' at the most important features of a face.

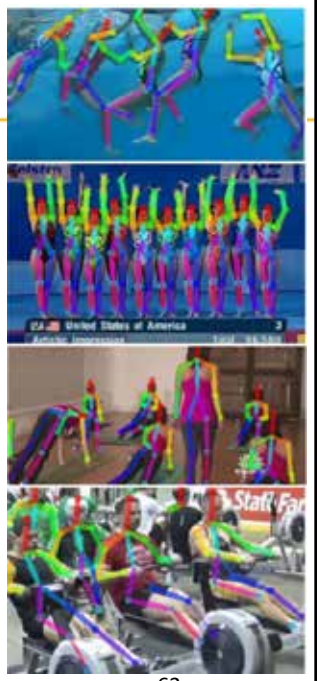


R. Ptucha '20 62

62

Human Pose

- Finding human pose body joints is difficult as we don't know how many people in the image, and even if we did, there can be complex interactions.
- Many methods first find all the people, then determine body pose of each person.
- These top down methods are reliant on accurate people detectors, and can be slow if many people in image.
- Bottom up approaches find all parts, and then try to assemble into bodies.

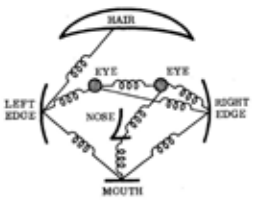


Cao et al. CVPR'17

R. Ptucha '20 63

63

Traditional Pose



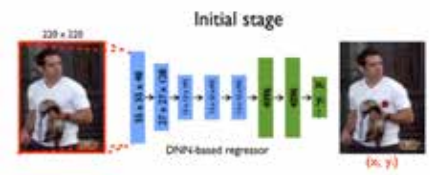
- Objects are represented as a collection of parts.
- Parts are connected in an ordered fashion.
- Deformable parts models use physical constraints, often with global and local representations.
- Difficult to model all situations and scenarios.

https://blog.nanonets.com/human-pose-estimation-2d-guide/

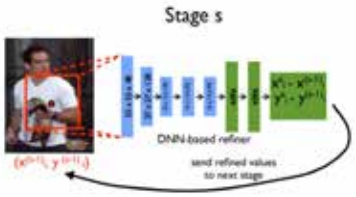
64

CNN Pose

- By replacing the last layer of a CNN with XY body joints and a L2 loss, supervised techniques leave all modelling to the nnet.



- DeepPose [Toshev and Szegedy '14] discovered that adding a second stage can refine initial coarse estimates by considering each joint's local neighborhood.



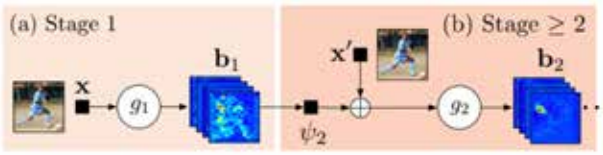
https://arxiv.org/pdf/1312.4659.pdf

65

Usage of Masks

- Follow-on work generated a heatmap for each joint.

Tompson et al. '15, https://arxiv.org/pdf/1411.4280.pdf



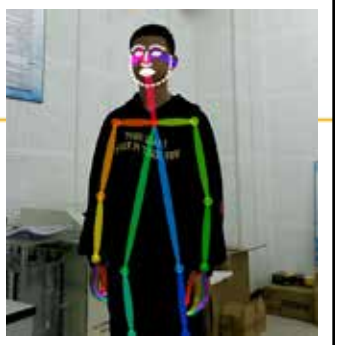
Wei et al. '16 https://arxiv.org/pdf/1602.00134.pdf

OpenPose

Cao et al. CVPR'17

https://github.com/CMU-Perceptual-Computing-Lab/openpose

- Real-time multi-person landmark detection.
- Outputs 2D joint/landmarks from regular webcams.
- 15 or 18 keypoint body joint estimation. [Cao et al. CVPR'17, Wei et al. CVPR'16]
- 21 keypoint hand/finger estimation. [Simon et al. CVPR'17]
- 70 landmark face estimation. [Simon et al. CVPR'17]

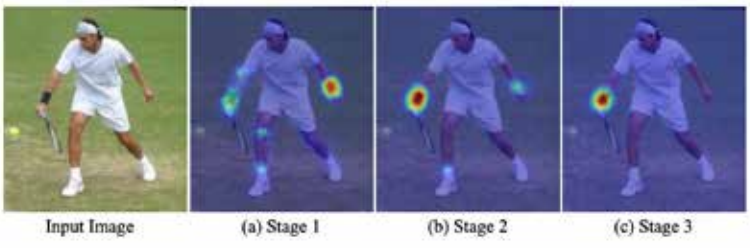


Yuan et al. FG2019

Convolution Pose Machines

https://arxiv.org/abs/1602.00134

- Developed at CMU, used in OpenPose.
- Sequential convolutional networks that directly operate on belief maps from previous stages, producing increasingly refined estimates for part locations.



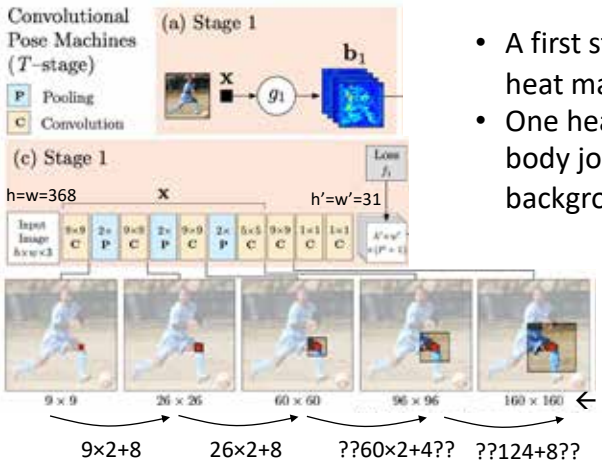
R. Ptucha '20

68

68

Convolution Pose Machines

https://arxiv.org/abs/1602.00134



- A first stage generates (*P*+1) heat maps.
- One heat map for each of *P* body joints, plus one for background.

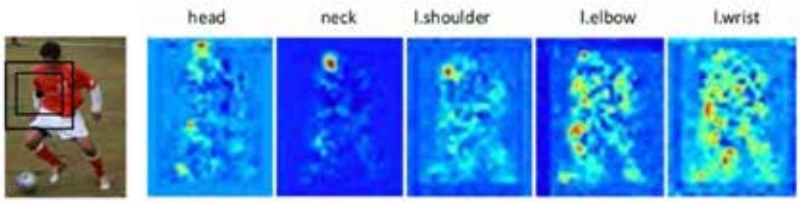
← Receptive field of each pixel

R. Ptucha '20

69

69

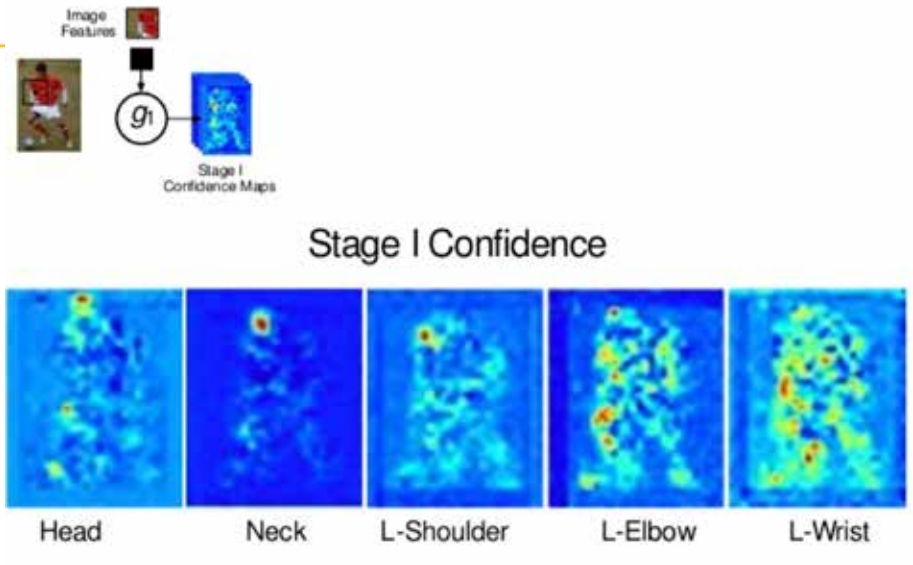
- Local evidence is weak



https://www.slideshare.net/plutoyang/pose-machine

R. Ptucha '20 70

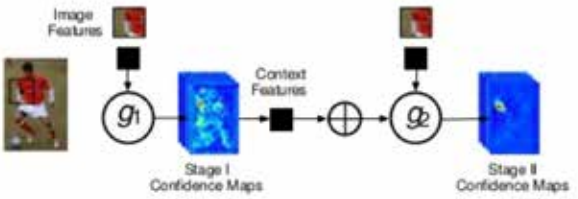
70



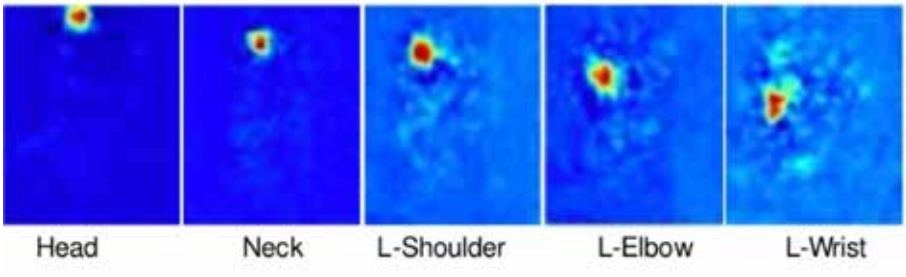
https://www.slideshare.net/plutoyang/pose-machine

R. Ptucha '20 71

71



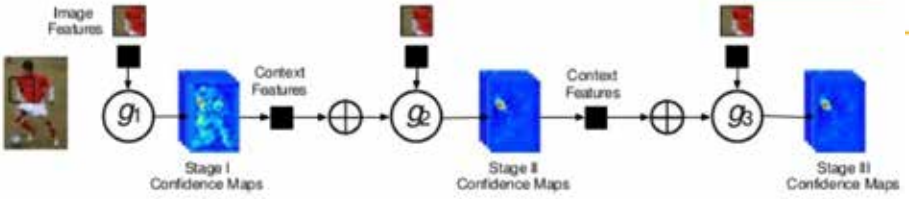
Stage II Confidence



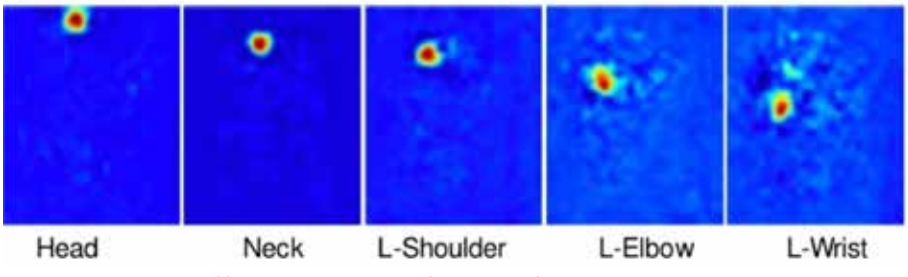
https://www.slideshare.net/plutoyang/pose-machine

R. Ptucha '20 72

72



Stage III Confidence



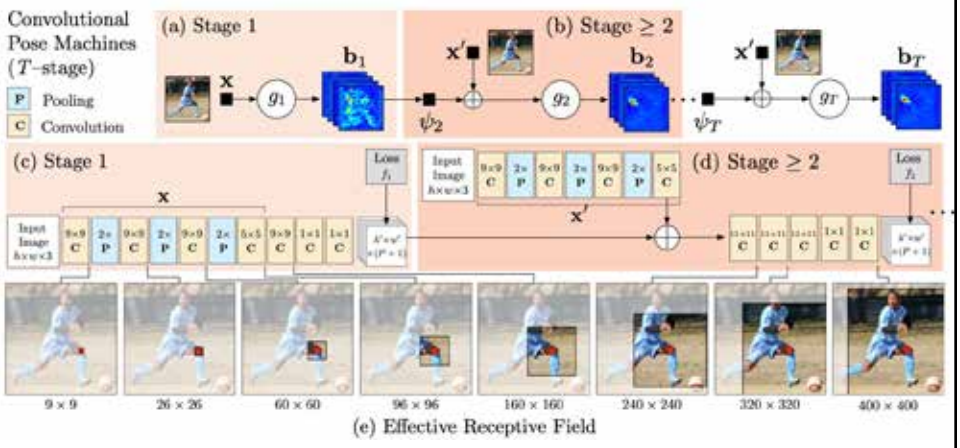
https://www.slideshare.net/plutoyang/pose-machine

R. Ptucha '20 73

73

Convolution Pose Machines

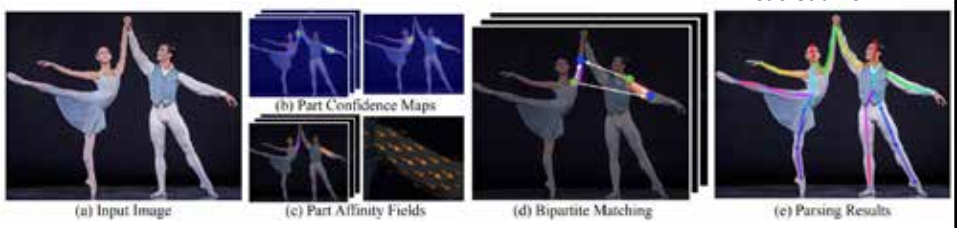
https://arxiv.org/abs/1602.00134



- First stage maps are concatenated with second stage maps

Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields

Cao et al. CVPR'17



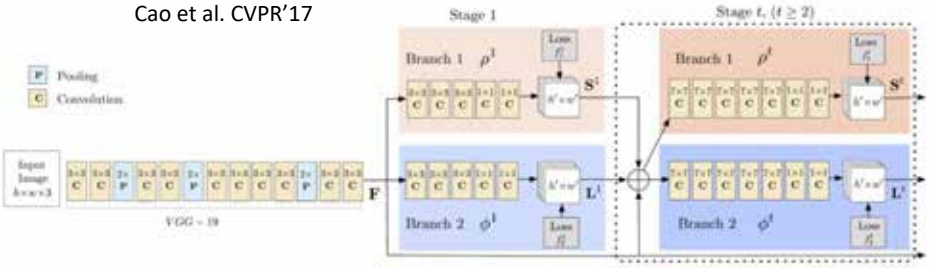
CNN simultaneously predicts a set of 2D confidence maps of body part locations (b), one map per body part;

and a set of 2D vector fields called Part Affinity Fields (PAF) (c) that encode the location and orientation of limbs, one map per body part.

Form all bipartite graphs, measure the alignment of the predicted PAF with the candidate limb that would be formed by connecting the detected body parts. Use a variation of the Hungarian algorithm to find the optimal graph.

Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields

Cao et al. CVPR'17



Stage 0: The first 10 layers of VGGNet are used to create feature maps for the input image.

Stage 1: Top branch predicts a set of 2D confidence maps (S) of body part locations (e.g. elbow, knee etc.). The second branch predicts a set of 2D vector fields (L) of part affinities, which encode the degree of association between parts.

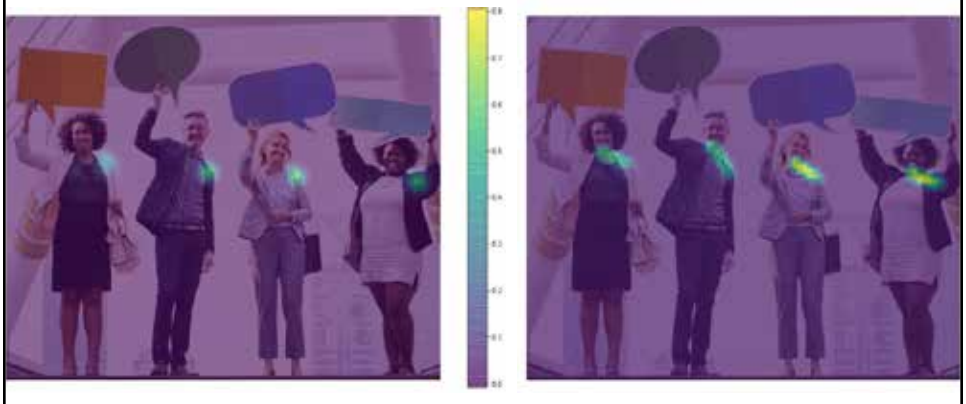
Stage 2: The confidence and affinity maps are parsed by greedy inference to produce the 2D keypoints.

76

Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields

Confidence maps for Left Shoulder

Confidence maps for Part Affinity maps for Neck – Left Shoulder

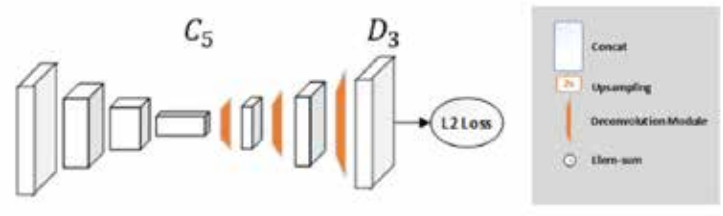


https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/

77

Simple Baselines for Human Pose Estimation and Tracking,

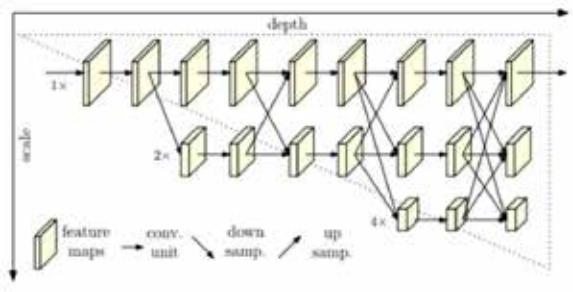
Xiao et al. ECCV '18, https://arxiv.org/pdf/1804.06208.pdf



- Stack a bunch of deconv layers onto a ResNet baseline.
- Simple and effective!

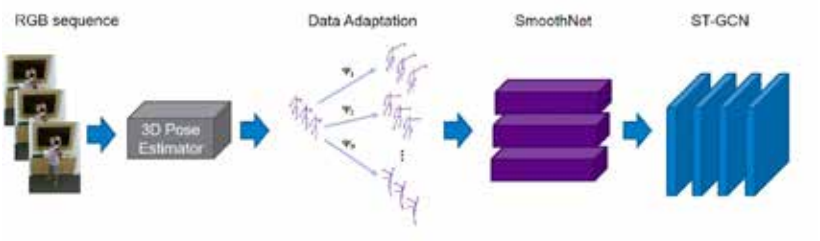
Deep High-Resolution Representation Learning for Human Pose Estimation,

Sun et al. CVPR'19 , https://arxiv.org/pdf/1902.09212.pdf

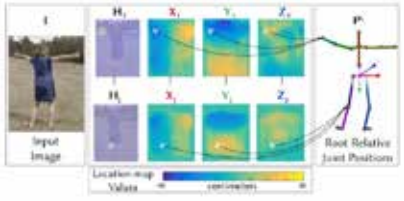


- Rather than use a Deconvnet (high to low back to high resolution) architecture, maintain high resolution to feed heat maps.

DeepVI: A Novel Framework for Learning Deep View-Invariant Human Action Representations using a Single RGB Camera [Papadopoulos et al. FG'20]



- No depth needed.
- Take advantage of temporal sequence



R. Ptucha '20 80

80

Detectron2

https://github.com/facebookresearch/detectron2

Pytorch: panoptic segmentation, densepose, Cascade R-CNN, rotated bounding boxes, etc

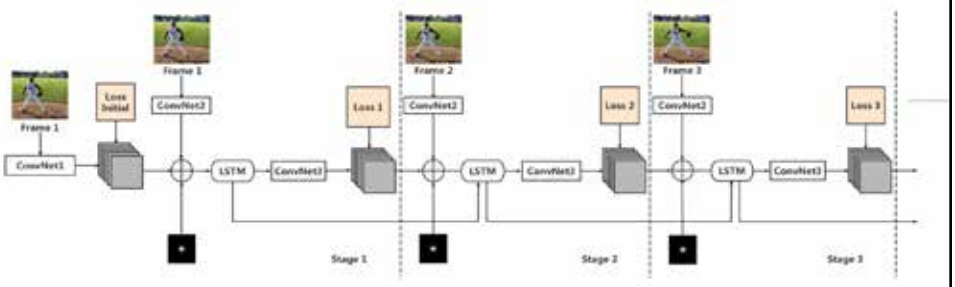
R. Ptucha '20 81

81

Pose From Video: LSTM Pose Machine

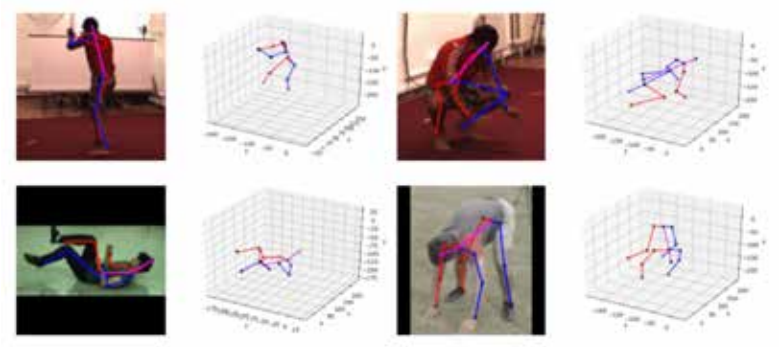
https://arxiv.org/abs/1712.06316

- Weight sharing for multi-stage CNN can be re-written as a Recurrent Neural Network.
- Enables the adoption of Long Short-Term Memory (LSTM) units between video frames.
- Can impose geometric consistency among frames.



3D Human Pose Estimation

- Estimate XY, and Z of each joint from RGB image.



https://blog.nanonets.com/human-pose-estimation-3d-guide/

3D Human Pose Estimation

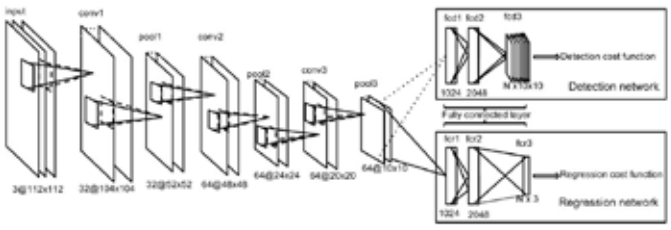
- Not only is this harder than XY joint estimation, it is difficult to collect ground truth data.

https://blog.nanonets.com/human-pose-estimation-3d-guide/

84

3D Human Pose Estimation

- Early deep approaches extended XY CNNs to XYZ CNNs.

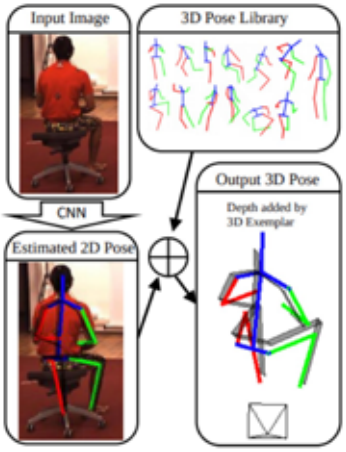


Li and Chan, ACCV'14 http://visal.cs.cityu.edu.hk/static/pubs/conf/accv14-3dposecnn.pdf

85

Merging 2D images with 3D Pose Libraries

- Although getting XYZ ground truth joint data from RGB images difficult, there are many 3D pose datasets.
- Two sources can be merged.

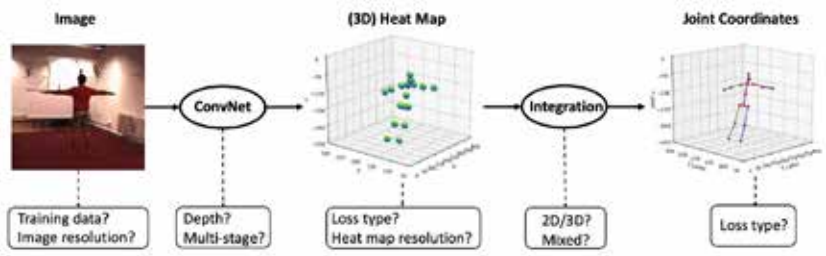


Chen and Ramanan, CVPR'17 https://arxiv.org/pdf/1612.06524.pdf

86

Integral Human Pose Regression

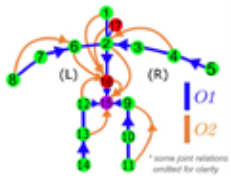
Sun et al., ECCV'18 https://arxiv.org/pdf/1711.08229.pdf



- Taking the max from a heat map is not differentiable, preventing end-to-end training.
- By modifying the taking maximum to taking the expectation operation, the final joint location is essentially a linear summation of all points in the mask,
- The method is called integral regression ,where each point on the mask has a corresponding ground truth for end to end training.

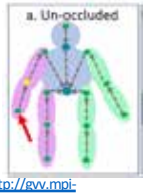
87

Website which tracks papers which do 3D post estimation:
https://paperswithcode.com/task/3d-human-pose-estimation



https://arxiv.org/pdf/1611.09813.pdf

- Estimate global positions of skeleton joints accounting for camera viewpoint.
- Regress position relative to root (joint 15), or relative to first (blue) and second order (orange) skeleton hierarchy, then add global position.



http://gvv.mpi-inf.mpg.de/projects/SingleShotMultiPerson/)

- To collect ground truth on real-world data, it is common to use multiple cameras capturing a scene (The Captury. http://www.thecaptury.com/, 2016.).
- For error analysis, two popular choices:
  1. Mean Per Joint Position Error (MPJPE)
  2. Use percentage of correct keypoints (PCK), then area under curve (AUC) for various thresholds. If use one threshold, treat the prediction as correct if it lies within a 15cm ball centered at the ground-truth joint location.
- Evaluate over 14 joints shown above.

88

LCR-NET++

(PAMI2019) https://arxiv.org/abs/1803.00455

N region proposals, each tested over *K* anchor poses

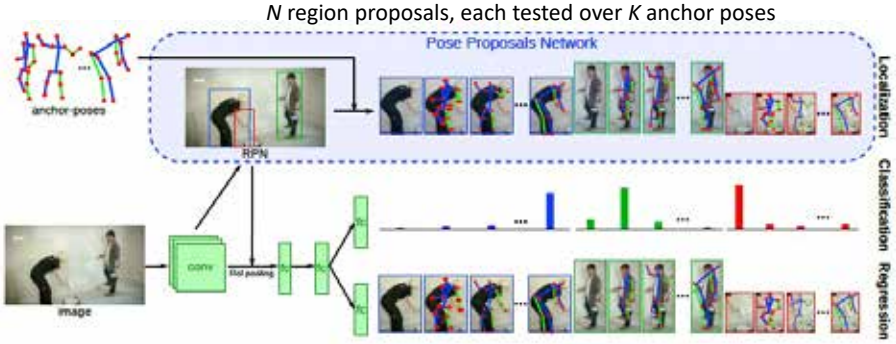


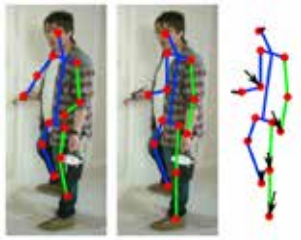
Fig. 2. Overview of our LCR-Net architecture (poses only shown in 2D for better readability). We first extract candidate regions using a Region Proposal Network (RPN) and obtain pose proposals by placing a fixed set of anchor-poses into these boxes (top). These pose proposals are then scored by a classification branch and refined using class-specific regressors, learned independently for each anchor-pose.

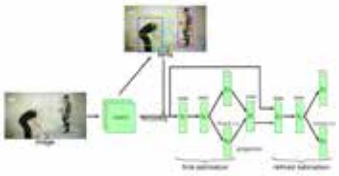
89

LCR-NET++

(PAMI2019) https://arxiv.org/abs/1803.00455

- Generation and scoring of a number of pose proposals per image, which allows 2D and 3D pose predictions of multiple people simultaneously.
- Does not require an approximate localization of the humans for initialization.
- Three main components:
  1. A pose proposal generator that suggests candidate poses;
  2. A classifier that scores the different pose proposals; and
  3. A regressor that refines pose proposals both in 2D and 3D.
- All three stages share the convolutional feature layers and are trained jointly.
- Recovers full-body 2D and 3D poses, hallucinating plausible body parts when the persons are partially occluded or truncated by the image boundary





R. Ptucha '20 90

90

Redaction

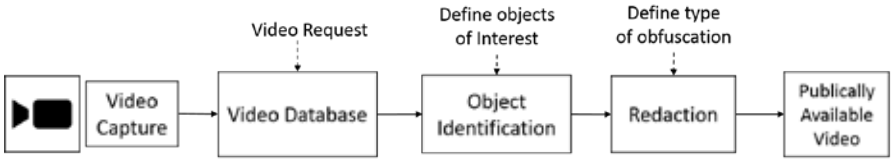
Sah et al. "Detection without Recognition for Redaction" 2017

- Redaction is used to obfuscate personally identifiable information such as faces, license plates, house numbers, and store fronts.
- Ideally, obfuscate targeted areas while maintaining scene context.
- Is there a facial obfuscation method that maintains context by allowing facial detection methods to find an obfuscated face, but prevent humans and facial recognition systems from identifying the same face?

R. Ptucha '20 91

91

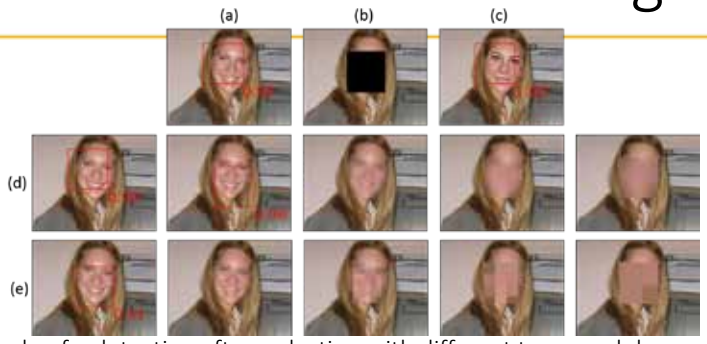
Video Redaction System



- MT-CNN [Zhang et al. 2016] for face detection
- OpenFace [Amos et al. 2016] implementation of FaceNet [Schroff et al. 2015] for face recognition

92

Detection without Recognition



Example of a detection after redaction with different types and degrees of obfuscations (red box drawn of face detected).

a) Original image
b) Detector fails when face obfuscated a mask
c) Face detected when original face swapped† with average female face
d) Varying degree of blurring using Gaussian kernel (sigma = 3, 5, 15, 25, 45)
e) Varying degree of pixelation (pixel block size = 2, 4, 8, 16, 32)

† http://matthewearl.github.io/2015/07/28/switching-eds-with-python/

93

Experimental Results

The effect of various obfuscation techniques on face detection

Face detection on AFLW [Koestinger et al. 2011] dataset. ~25K annotated faces with 21 landmarks .

Input		Detection (IoU)
Original image		0.688
Obfuscation	Full blur ($\sigma = 25$)	0.368
	Blur face ($\sigma = 5$)	0.713
	Blur face ($\sigma = 15$)	0.609
	Blur face ($\sigma = 25$)	0.421
	Blur face ($\sigma = 45$)	0.093
	Pixelate face ($p = 2$)	0.708
	Pixelate face ($p = 4$)	0.645
	Pixelate face ($p = 8$)	0.171
	Pixelate face ($p = 16$)	0.0
	Mask face	0.0
	Face swap	0.544

R. Ptucha '20

94

94

Experimental Results

Demonstrate the trade-off between face detection and recognition accuracies with varying degree of obfuscation.

Face detection and recognition on FaceScrub [Ng and Winkler '14] dataset. ~100K faces from 530 people

Input	Det. (IoU)	Recog. (% acc.)
Original image	0.901	98.29
Full blur ($\sigma = 25$)	0.570	48.80
Blur face ($\sigma = 5$)	0.898	97.53
Blur face ($\sigma = 15$)	0.893	91.11
Blur face ($\sigma = 25$)	0.769	68.32
Blur face ($\sigma = 45$)	0.228	7.66
Pixelate face ($p = 2$)	0.899	97.87
Pixelate face ($p = 4$)	0.889	96.10
Pixelate face ($p = 8$)	0.450	22.27
Pixelate face ($p = 16$)	0.000	0.02
Mask face	0.000	0.00
Face swap	0.903	34.73

Want High Want Low

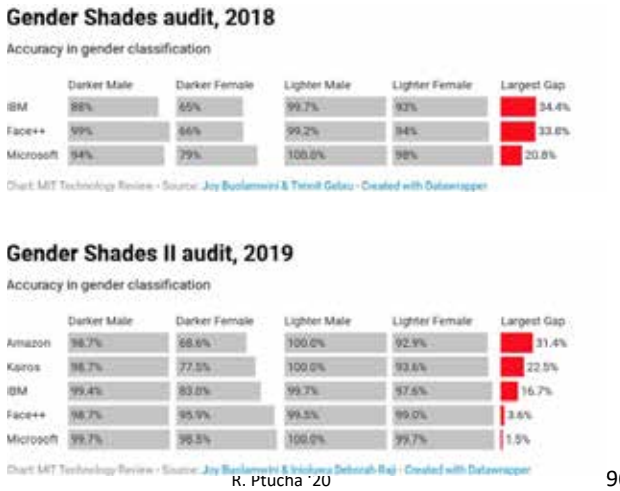
R. Ptucha '20

95

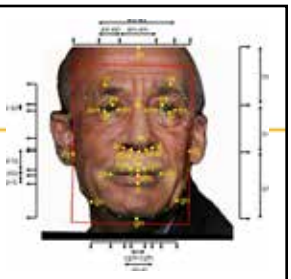
95

Unwanted Bias

http://www.aies-conference.com/wp-content/uploads/2019/01/AIES-19_paper_223.pdf



Diversity in Faces

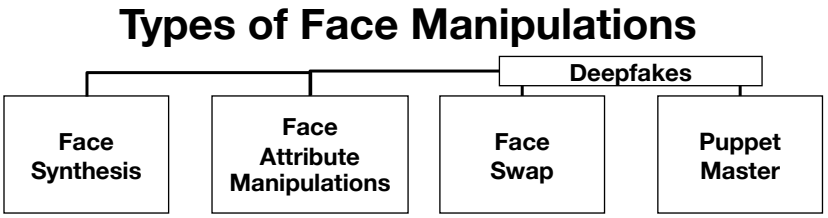


https://arxiv.org/pdf/1901.10436.pdf

- 1M facial dataset
- Ten generic facial coding schemes including things such as facial dimensions, age, skin color, contrast, gender.
- Emphasizes the inclusion of all races and the minimization of bias.
- Aims to make facial recognition systems of the future both more fair and accurate.

Deepfakes

- Deepfake are artificially generated audo-visual renderings of an individual.
- These audiovisual renderings can be used to defame a public figure or influence public opinion.

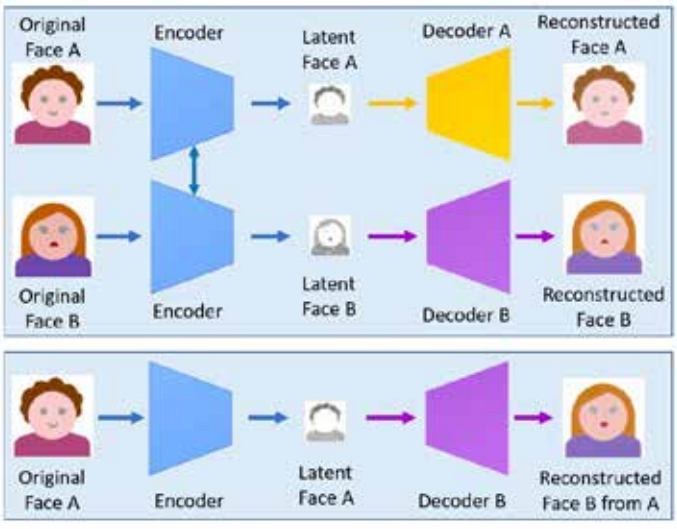


R. Ptucha '20 98

98

Deepfakes

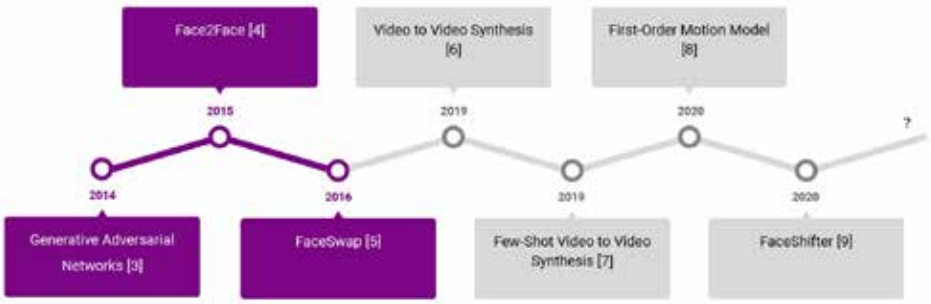
Deep Learning for Deepfakes Creation and Detection, Nguyen et al., 2020



R. Ptucha '20 99

99

Deepfakes- Generation



3. https://arxiv.org/abs/1406.2661
4. https://niessnerlab.org/papers/2019/8facetoface/thies2018face.pdf
5. https://github.com/deepfakes/faceswap
6. https://arxiv.org/abs/1808.06601
7. https://arxiv.org/abs/1910.12713
8. https://arxiv.org/abs/2003.00196
9. https://arxiv.org/abs/1912.13457
10. https://arxiv.org/abs/1809.00888

Face Swap

https://beebom.com/best-face-swap-apps/

- Apps: Snapchat, FakeApp, REFACE, Parodist
- Repos: Faceswap-GAN, Deepfakes,
- Swap identities between source and target faces

Different Types of Deepfakes

1. Lip-sync: A persons head and face are real, but the mouth being replaced (Peele's Obama video)
2. Face swap: A persons face is swapped onto another body.
3. Puppet master: Videos in which a persons body is digitized and made into a puppet that can be manipulated by an actor

Face2face, Thies et al.

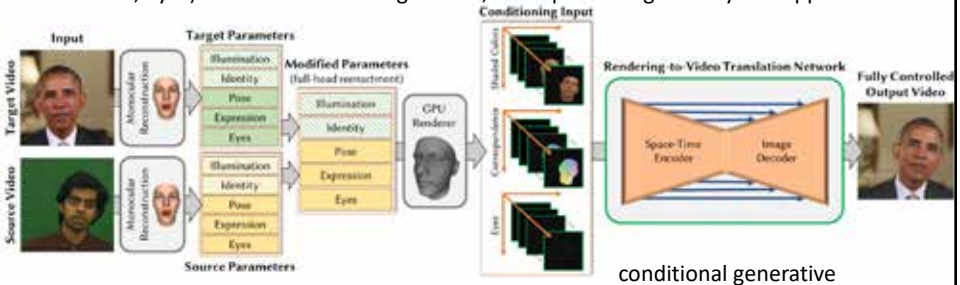
R. Ptucha '20 102

102

Face Reenactment

(Note can do same for voice- called voice transformation)

Learn statistical properties of target and source. Source head (pose, expression, mouth, eyes) takes full control target head, while preserving identity and appearance.



Extract parametric face information

Space-time encodings in a sliding window which can keep illumination and Identity, replace pose and expression.

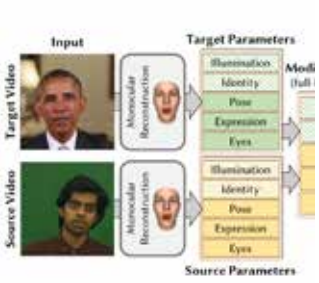
conditional generative adversarial network (cGAN) [Isola et al. CVPR'17], modified for consistent face, hair and upper body over video.

https://web.stanford.edu/~zollhoef/papers/SG2018_DeepVideo/paper.pdf

R. Ptucha '20 103

103

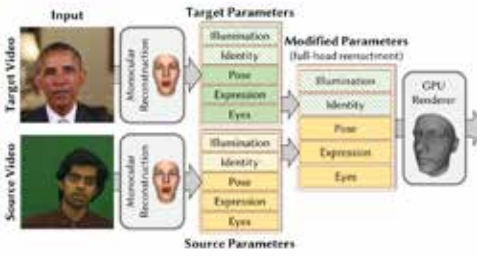
Face Reenactment



For each frame of target and source $\in \mathbb{R}^{261}$, extract:

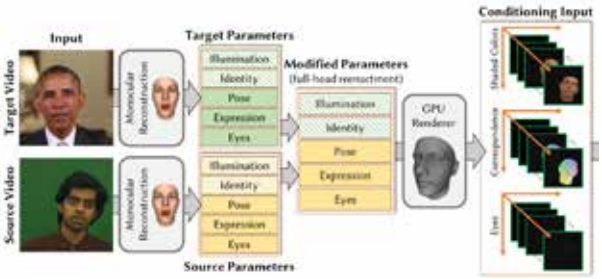
- Facial pose $\in \mathbb{R}^3$
- Translation $\in \mathbb{R}^3$
- Linear combination of PCA basis functions:
  - Facial identity, geometry $\in \mathbb{R}^{80}$
  - Facial identity, reflectance $\in \mathbb{R}^{80}$
  - Facial expression, i $\in \mathbb{R}^{64}$
- Gaze direction for eyes $\in \mathbb{R}^4$
- Incoming illumination $\in \mathbb{R}^{27}$

Face Reenactment



Reconstruct new dense face using illumination and identity of target and facial pose/expression from source.

Face Reenactment



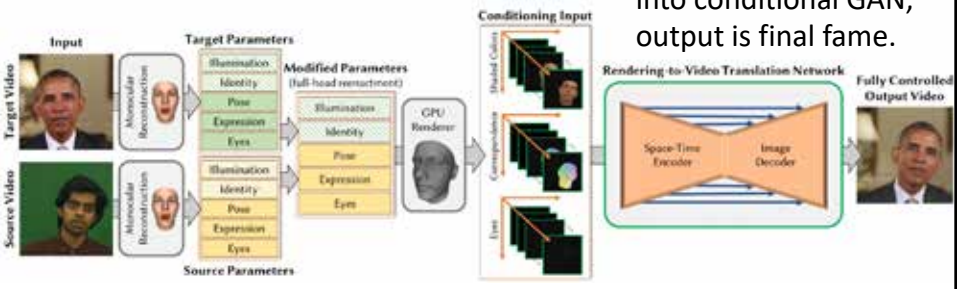
Synthetic face provides good starting point for next stage.
For each window of three frames:

1. Color rendering of new face.
2. Correspondence image for texture.
3. Eye blink and gaze control.

R. Ptucha '20 106

106

Face Reenactment



Three frames, each with three input passed into conditional GAN, output is final fame.

Synthesize face, hair, body, and background
One model for each target actor and static background

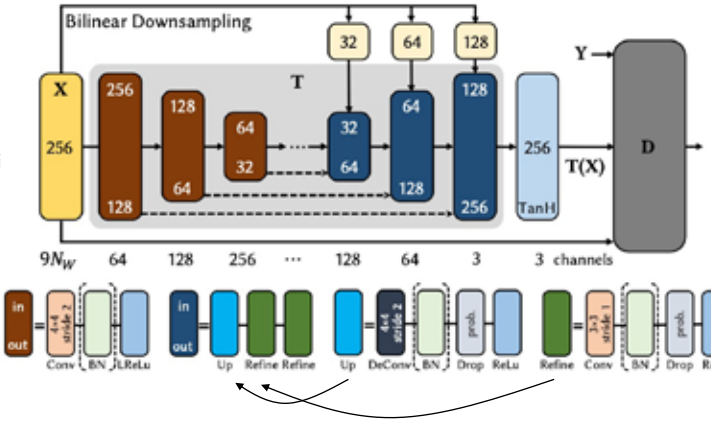
R. Ptucha '20 107

107

Face Reenactment

Rendering to video architecture

Input 256x256x 3famesx3 conditioni mages

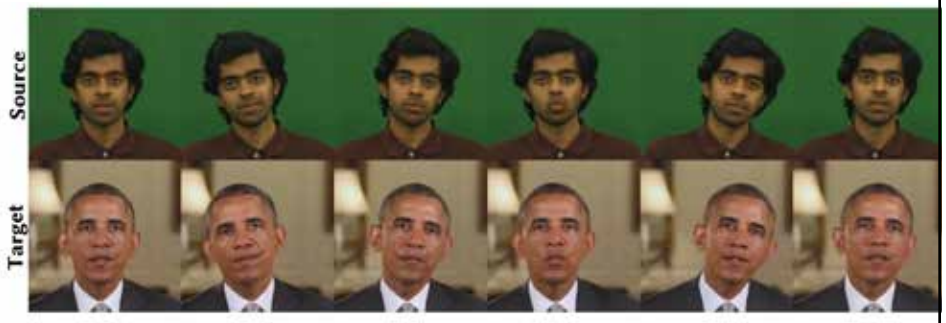


Encoder has 8 downwampling modules from 256x256 down to 2x2, decocer has 8 upsampling modules back up to 256x256.

R. Ptucha '20 108

108

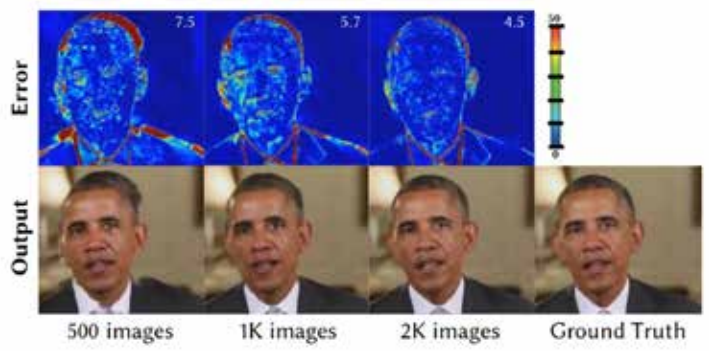
Face Reenactment



R. Ptucha '20 109

109

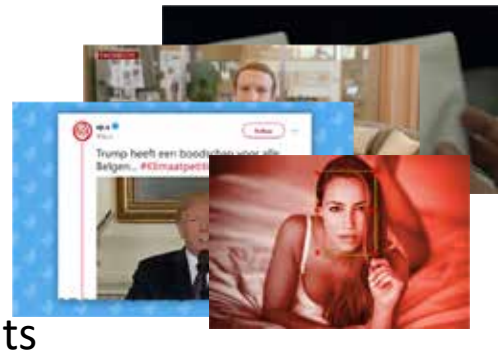
Face Reenactment
How Many Target Images Needed for Training?



110

Dangers of Deepfakes

- Rapid research to implementation
- Potential misuses
  - Propaganda
  - Blackmail
  - Bullying, etc
- Rapid improvements



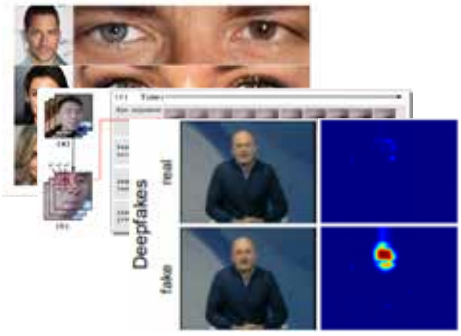
111

Defense Against Deepfakes

Biological Signals

Pixel Abnormalities

State-of-the-art is primarily deep learning based



FaceForensics++
Rossler et al.

112

Defense Against Deepfakes

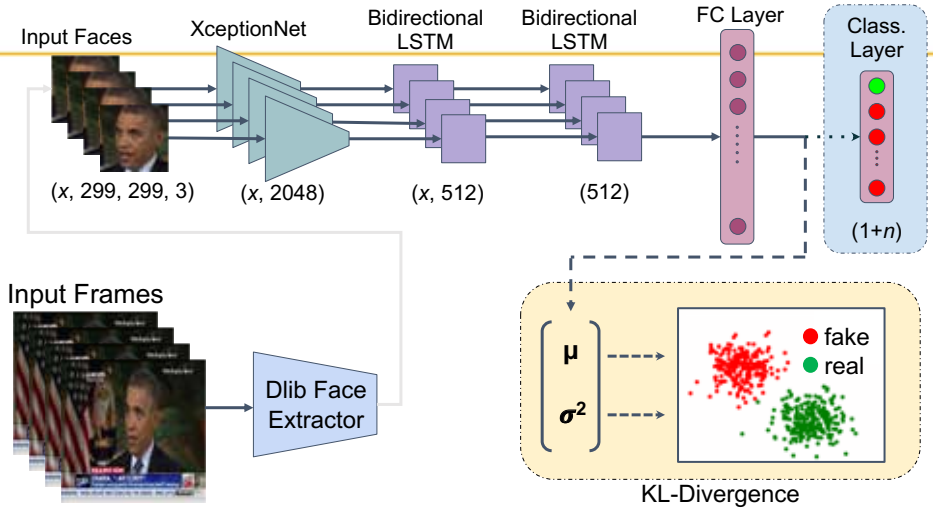
Deepfake Detector

Real/Fake

Spoof Detector

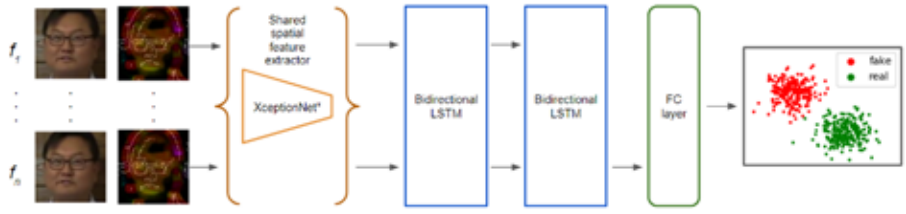
Real/Fake

113



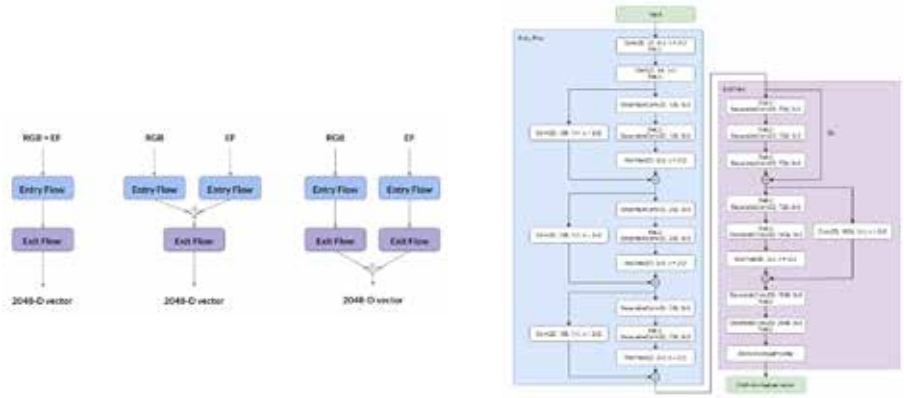
Chintha, et al. "Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection." IEEE Journal of Selected Topics in Signal Processing (2020).

Using Edges and Optical Flow



Chintha, et al. "Leveraging edges and optical flow on faces for deepfake detection" IJCB (2020).

Using Edges and Optical Flow – Model Variants



Chintha, et al. "Leveraging edges and optical flow on faces for deepfake detection" IJCB (2020).

Thank you!!

Ray Ptucha
rwpeec@rit.edu

https://www.rit.edu/mil