Deep Learning for Computer Vision

CNNs for Human Understanding: Human Pose and Crowds

Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Human Pose Estimation

- Problem of localization of human joints (also known as keypoints - elbows, wrists, etc) in images or videos

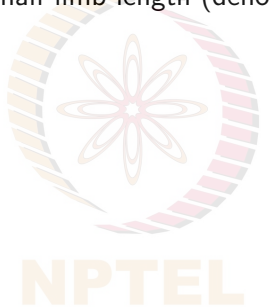
Human Pose Estimation

Credit: Bearman et al, Toshev et al

- Problem of localization of human joints (also known as keypoints - elbows, wrists, etc) in images or videos
- HPE task pipelines broadly classified into:
  - **Single-Person Pipeline:** Regression-based, Detection-Based
  - **Multi-Person Pipeline:** Top-down, Bottom-up approaches

HPE: How to evaluate?

- **Percentage of Correct Parts (PCP):** A limb is considered detected if distance between detected joint and true joint $<$ half limb length (denoted as PCP@0.5)



Credit: A 2019 Guide to Human Pose Estimation with Deep Learning by Nanonets

HPE: How to evaluate?

- **Percentage of Correct Parts (PCP):** A limb is considered detected if distance between detected joint and true joint $<$ half limb length (denoted as PCP@0.5)
- **Percentage of Detected Joints (PDJ)**: A detected joint is correct if distance between predicted and true joint is within certain fraction of torso diameter; e.g. PDJ@0.2 $\implies$ distance between predicted and true joint $< 0.2\times$ torso diameter

Credit: A 2019 Guide to Human Pose Estimation with Deep Learning by Nanonets

HPE: How to evaluate?

- **Percentage of Correct Parts (PCP):** A limb is considered detected if distance between detected joint and true joint $<$ half limb length (denoted as PCP@0.5)
- **Percentage of Detected Joints (PDJ)**: A detected joint is correct if distance between predicted and true joint is within certain fraction of torso diameter; e.g. PDJ@0.2 $\implies$ distance between predicted and true joint $< 0.2\times$ torso diameter
- **Object Keypoint Similarity (OKS) based mAP:**

$$OKS = \frac{\sum_i \exp(\frac{-d_i^2}{2s^2k_i^2})\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}$$

where d_i is Euclidean distance between detected keypoint and corresponding ground truth, v_i is visibility flag of ground truth, s is object scale, and k is per-keypoint constant that controls falloff (OKS is IoU equivalent for keypoint evaluation)

Credit: A 2019 Guide to Human Pose Estimation with Deep Learning by Nanonets

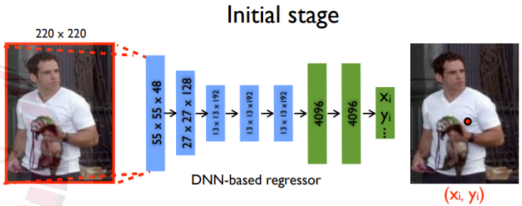
Regression-based Methods: DeepPose[1]

- First work to kick off deep learning-based HPE

[1] Toshev et al, DeepPose: Human Pose Estimation via Deep Neural Networks, CVPR 2014

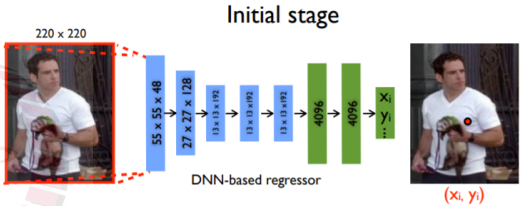
Regression-based Methods: DeepPose[1]

- First work to kick off deep learning-based HPE
  - **Model:** AlexNet-inspired
  - **I/O:** Input image; Output is $\mathbf{y} = (..., \mathbf{y}_i^\mathsf{T}, ...)^\mathsf{T}$ where $\mathbf{y}_i$ contains x and y coordinates of i^{th} joint
  - **Loss:** L_2-norm for regression



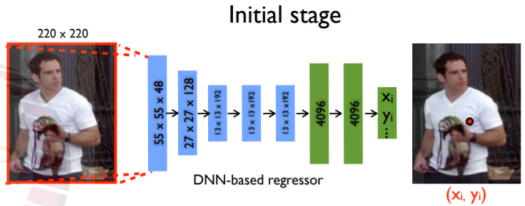
Regression-based Methods: DeepPose[1]

- First work to kick off deep learning-based HPE
  - **Model:** AlexNet-inspired
  - **I/O:** Input image; Output is $\mathbf{y} = (..., \mathbf{y}_i^\mathsf{T}, ...)^\mathsf{T}$ where $\mathbf{y}_i$ contains x and y coordinates of i^{th} joint
  - **Loss:** L_2-norm for regression
- Predictions are refined using **Cascaded Regressors**



Regression-based Methods: DeepPose[1]

- First work to kick off deep learning-based HPE
  - **Model:** AlexNet-inspired
  - **I/O:** Input image; Output is $\mathbf{y} = (..., \mathbf{y}_i^\mathsf{T}, ...)^\mathsf{T}$ where $\mathbf{y}_i$ contains x and y coordinates of i^{th} joint
  - **Loss:** L_2-norm for regression
- Predictions are refined using **Cascaded Regressors**
- Cropped images along with predicted joints are fed to network in next stages
- Forces model to learn generic features across finer image scales leading to high precision



[1] Toshev et al, DeepPose: Human Pose Estimation via Deep Neural Networks, CVPR 2014

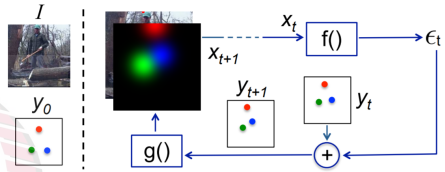
Regression-based Methods: Iterative Error Feedback[2]

- Mean pose recursively updated to match ground truth

[2] Carreira et al, Human Pose Estimation with Iterative Error Feedback, CVPR 2016

Regression-based Methods: Iterative Error Feedback[2]

- Mean pose recursively updated to match ground truth
- Given image concatenated with output representation, f is trained to predict "correction" that brings mean poses closer to ground truth

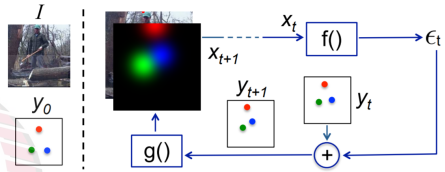


[2]Carreira et al, Human Pose Estimation with Iterative Error Feedback, CVPR 2016

Regression-based Methods: Iterative Error Feedback[2]

- Mean pose recursively updated to match ground truth
- Given image concatenated with output representation, f is trained to predict "correction" that brings mean poses closer to ground truth
- Mathematically:

$$
\begin{aligned}
x_0 &= I \\
\epsilon_t &= f(x_t) \\
y_{t+1} &= y_t + \epsilon_t \\
x_{t+1} &= x_t \oplus g(y_{t+1})
\end{aligned}
$$

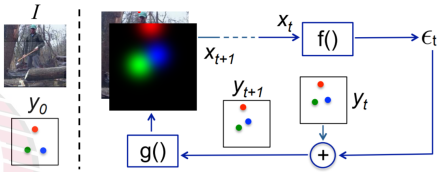


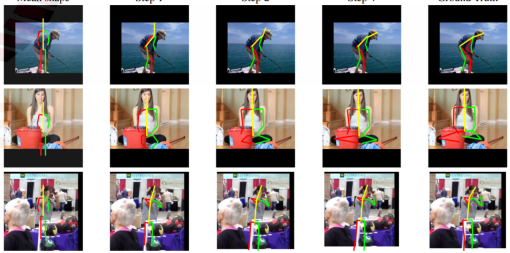
[2] Carreira et al, Human Pose Estimation with Iterative Error Feedback, CVPR 2016

Regression-based Methods: Iterative Error Feedback[2]

- Mean pose recursively updated to match ground truth
- Given image concatenated with output representation, f is trained to predict "correction" that brings mean poses closer to ground truth
- Mathematically:

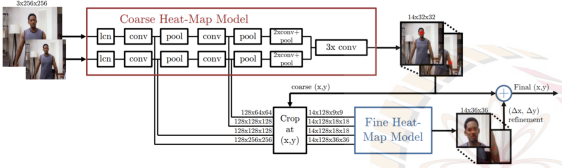
$$
\begin{aligned}
x_0 &= I \\
\epsilon_t &= f(x_t) \\
y_{t+1} &= y_t + \epsilon_t \\
x_{t+1} &= x_t \oplus g(y_{t+1})
\end{aligned}
$$



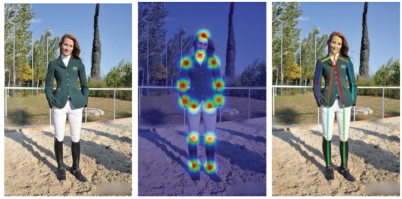


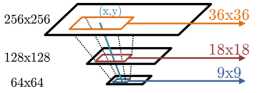
[2]Carreira et al, Human Pose Estimation with Iterative Error Feedback, CVPR 2016

Detection-based Methods[3]

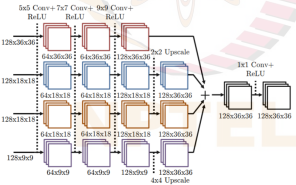


Overview of Cascaded Architecture





Crop module functionality for a single joint



The fine heat-map network for a single joint

Recover spatial accuracy lost due to pooling of model by using additional ConvNet to refine localization result of coarse heat-map

[3] Tompson et al, Efficient Object Localization using Convolutional Networks, CVPR 2015

Multi-Person Pose Estimation: Top-Down Pipeline

Multi-Person Pose Estimation: Top-Down Pipeline

- Detect all persons from given image
- Single-person approaches performed in each detected bounding box
- Context information from whole image can be used to improve performance

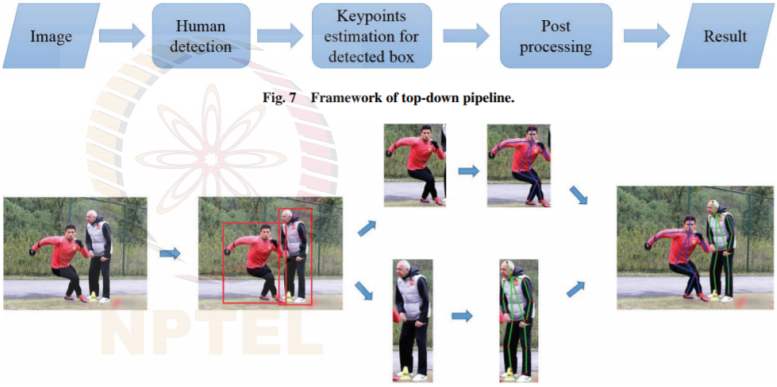


Fig. 7 Framework of top-down pipeline.

Fig. 8 An illustration of top down pipeline. (a) Input image, (b) two persons detected by human detector, (c) cropped single person image, (d) single person pose detection result, and (e) multi-person pose detection result.

Credit: Dang et al, Deep Learning based 2D Human Pose Estimation: A Survey, 2019

Multi-Person Pose Estimation: Bottom-Up Pipeline

- Procedure reversed from top-down
- All body parts (keypoints) are detected in first stage, then associated to human instances in second stage
- Inference stage likely to be faster - since no need to detect pose for each person separately

Fig. 10 Framework of bottom-up pipeline.

Fig. 11 An illustration of bottom-up pipeline. (a) Input image, (b) keypoints of all the person, and (c) all detected keypoints are connected to form human instance.

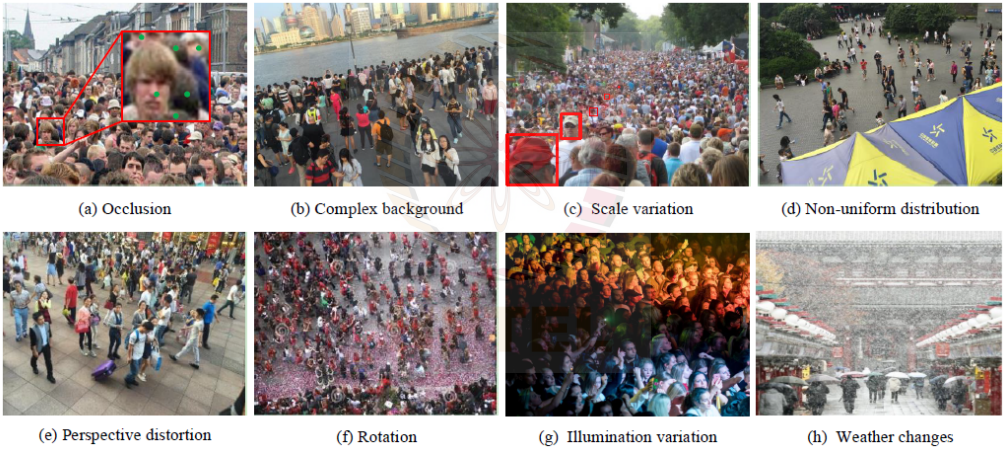
Credit: Dang et al, Deep Learning based 2D Human Pose Estimation: A Survey, 2019

Crowd Counting

Estimating crowd density in images a crucial task for urban planning, public safety and security

Credit: ShanghaiTech Dataset (Analytics Vidhya)

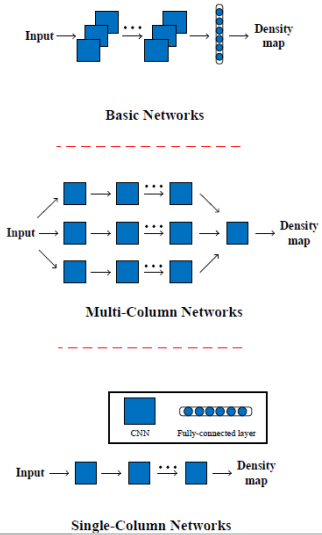
Crowd Counting: Why is it hard?



Credit: Gao et al, CNN-based Density Estimation and Crowd Counting: A Survey, 2020

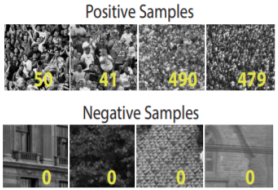
CNNs for Crowd Counting

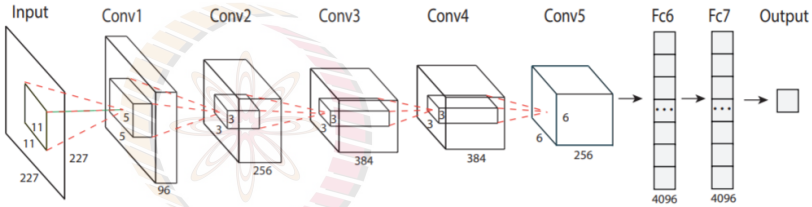
- **Basic CNN:** Basic CNN layers with no additional feature information
- **Multi-column:** Usually adopt different columns to capture multi-scale information corresponding to different receptive fields
- **Single-column:** Usually deploy single and deeper CNNs; premise to not increase complexity of network



Credit: Gao et al, CNN-based Density Estimation and Crowd Counting: A Survey, 2020

Basic CNN Approach[4]

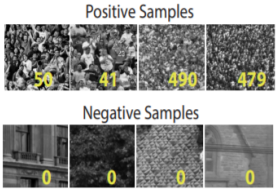


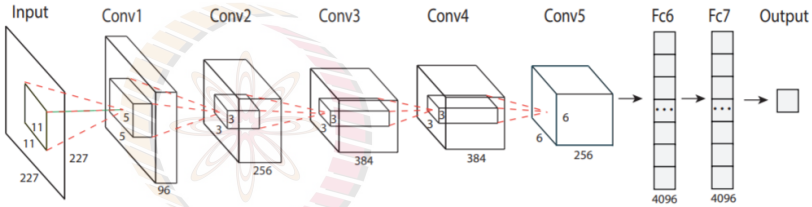


- One of first efforts to use CNNs for direct regression; based on AlexNet architecture for dense crowd counting

[4]Wang et al, Deep People Counting in Extremely Dense Crowds, ACM MM 2015

Basic CNN Approach[4]

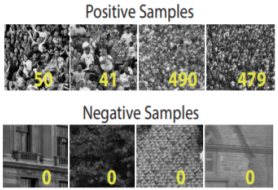


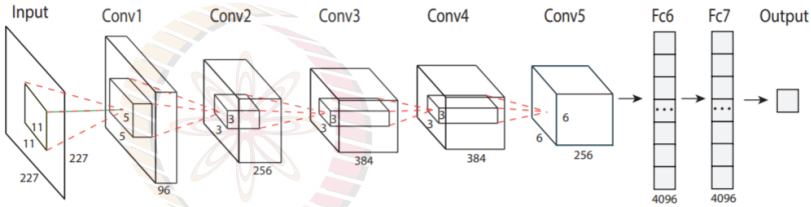


- One of first efforts to use CNNs for direct regression; based on AlexNet architecture for dense crowd counting
- Expanded set of negative samples, whose ground truth counts are zeros, used to reduce interference

[4]Wang et al, Deep People Counting in Extremely Dense Crowds, ACM MM 2015

Basic CNN Approach[4]



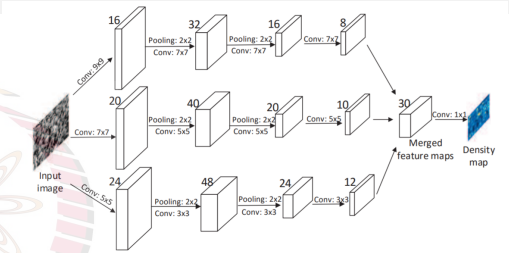


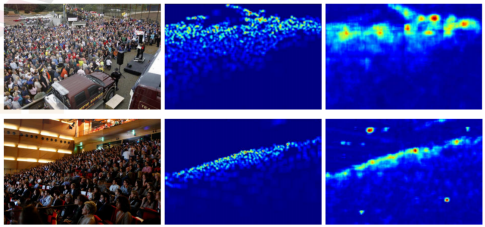
- One of first efforts to use CNNs for direct regression; based on AlexNet architecture for dense crowd counting
- Expanded set of negative samples, whose ground truth counts are zeros, used to reduce interference
- Sensitive to density, distribution of crowd and scale of people

[4]Wang et al, Deep People Counting in Extremely Dense Crowds, ACM MM 2015

Multi-column CNN

- Focused on multi-scale problem in crowd counting
- **Multi-column architecture:** Features learned by each column CNN adaptive to large variation in people/head size due to perspective effect or across different image resolutions
- Replaced fully connected layer with convolution layer whose filter size is 1×1; input image can be of arbitrary size to avoid distortion



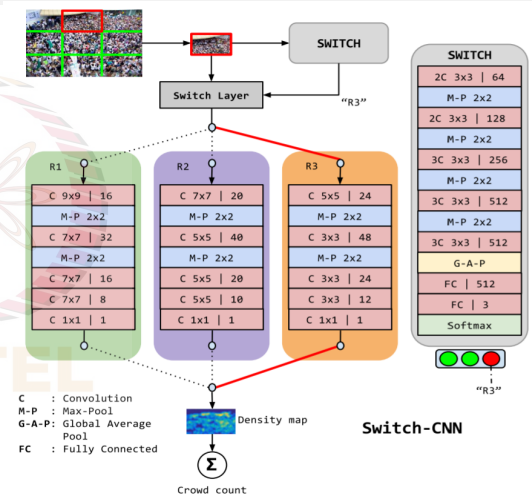


Test image Ground-truth Estimation

Zhang et al, Single-Image Crowd Counting via Multi-Column Convolutional Neural Network, CVPR 2016

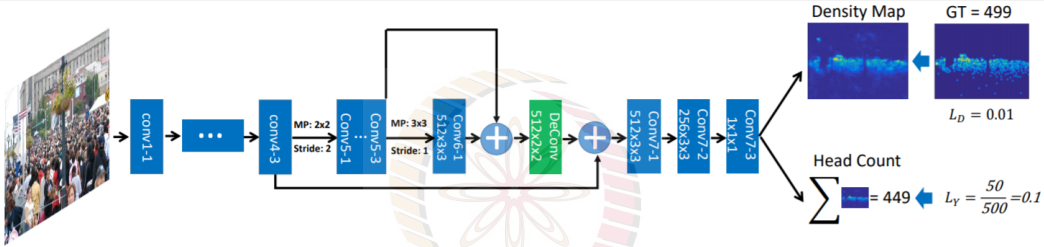
Multi-column CNN[5]

- Trains several independent CNN crowd density regressors on image patches (each regressor same as previous method of Zhang et al)
- **Switch classifier** trained alternatively on regressions to select best one for density estimation $\implies$ offers ability to model large-scale variations and leverage local variations in density in crowd scene
- Weighted averaging used to fuse features is global in nature



[5]Sam et al, Switching Convolutional Neural Network for Crowd Counting, CVPR 2017

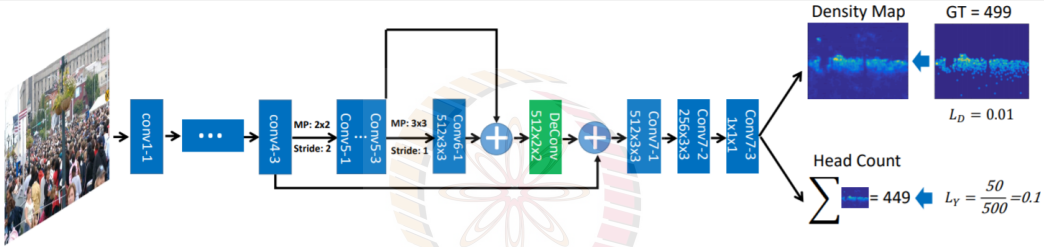
Single-column CNN[6]



- Based on observation that using a single column from multi-column networks retained 70% of accuracy on some datasets; hence, used a single-column CNN with single filter size as backbone

[6] Zhang et al, Crowd Counting via Scale-Adaptive Convolutional Neural Network, WACV 2018

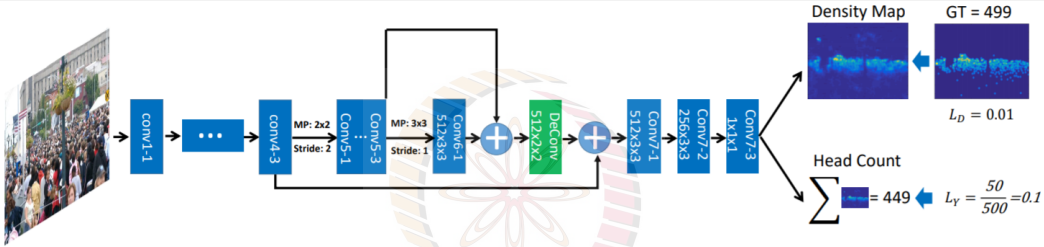
Single-column CNN[6]



- Based on observation that using a single column from multi-column networks retained 70% of accuracy on some datasets; hence, used a single-column CNN with single filter size as backbone
- By combining feature maps of multiple layers, could adapt network to variations in pedestrian (head) scale and perspective

[6] Zhang et al, Crowd Counting via Scale-Adaptive Convolutional Neural Network, WACV 2018

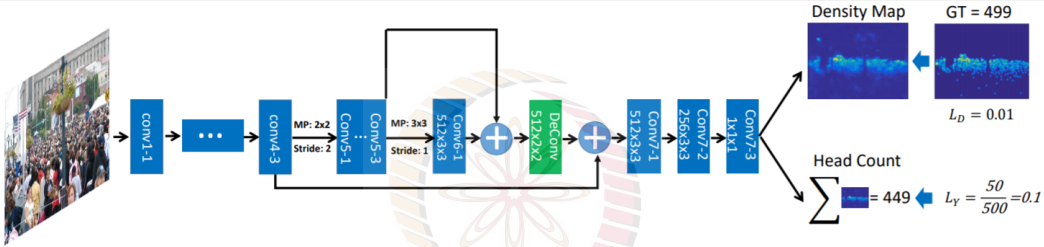
Single-column CNN[6]



- Based on observation that using a single column from multi-column networks retained 70% of accuracy on some datasets; hence, used a single-column CNN with single filter size as backbone
- By combining feature maps of multiple layers, could adapt network to variations in pedestrian (head) scale and perspective
- Used deconv layer to adapt network output instead of upsampling/elementwise summation

[6] Zhang et al, Crowd Counting via Scale-Adaptive Convolutional Neural Network, WACV 2018

Single-column CNN[6]

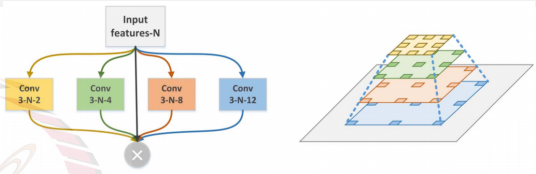


- Based on observation that using a single column from multi-column networks retained 70% of accuracy on some datasets; hence, used a single-column CNN with single filter size as backbone
- By combining feature maps of multiple layers, could adapt network to variations in pedestrian (head) scale and perspective
- Used deconv layer to adapt network output instead of upsampling/elementwise summation
- Entire network is optimized for density map estimation as well as for head count estimate

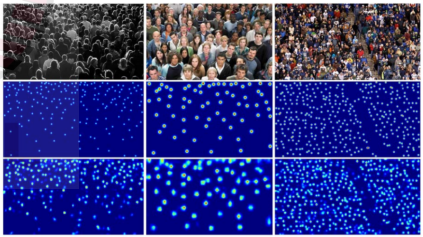
[6] Zhang et al, Crowd Counting via Scale-Adaptive Convolutional Neural Network, WACV 2018

Another Single-column CNN[7]

- Observed that low-level features from same depth of different columns in multi-column CNNs are similar



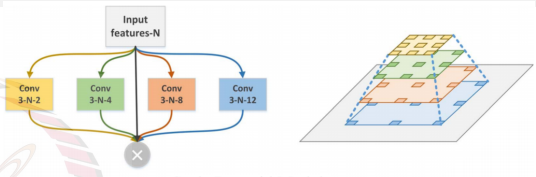
Scale Pyramid Module



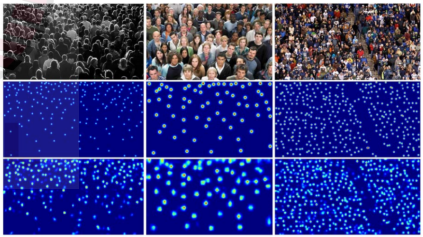
[7] Chen et al, Scale Pyramid Network for Crowd Counting, WACV 2019

Another Single-column CNN[7]

- Observed that low-level features from same depth of different columns in multi-column CNNs are similar
- Employ a single-column structure as shared backbone and extract multi-scale features from high-level features in high layers



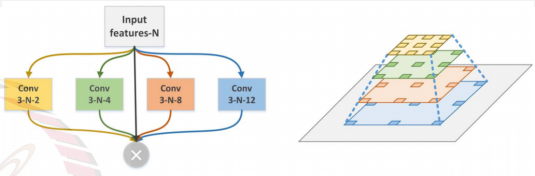
Scale Pyramid Module



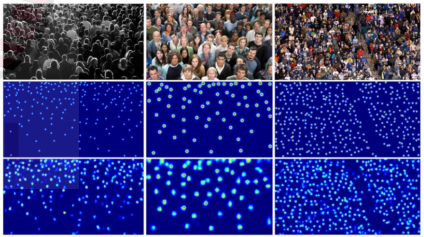
[7] Chen et al, Scale Pyramid Network for Crowd Counting, WACV 2019

Another Single-column CNN[7]

- Observed that low-level features from same depth of different columns in multi-column CNNs are similar
- Employ a single-column structure as shared backbone and extract multi-scale features from high-level features in high layers
- Use dilated convolutions which can obtain different receptive fields at different rates; this Scale Pyramid Module placed between Conv4_3 and Conv5_1 of VGG16



Scale Pyramid Module



[7] Chen et al, Scale Pyramid Network for Crowd Counting, WACV 2019

Homework

Readings

- A detailed blog post on human pose estimation by Nanonets
- Gao et al, CNN-based Density Estimation and Crowd Counting: A Survey, 2020
- (Optional) Dang et al, Deep Learning Based 2D Human Pose Estimation: A Survey, 2019

Exercise

CNNs for human understanding can especially suffer from biases in datasets (towards a particular race, ethnic background or gender); how do you find if a model is biased?

References I

[1] Alexander Toshev and Christian Szegedy. "DeepPose: Human Pose Estimation via Deep Neural Networks". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1653–1660.

[2] Amy L. Bearman, Stanford, and Catherine Dong. "Human Pose Estimation and Activity Classification Using Convolutional Neural Networks". In: 2015.

[3] Jonathan Tompson et al. "Efficient object localization using Convolutional Networks". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 648–656.

[4] Chuan Wang et al. "Deep People Counting in Extremely Dense Crowds". In: Oct. 2015, pp. 1299–1302.

[5] João Carreira et al. "Human Pose Estimation with Iterative Error Feedback". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4733–4742.

[6] Yingying Zhang et al. "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 589–597.

[7] Deepak Sam, Shiv Surya, and R. Babu. "Switching Convolutional Neural Network for Crowd Counting". In: July 2017, pp. 4031–4039.

References II

[8] Lu Zhang, Miaojing Shi, and Qiaobo Chen. "Crowd Counting via Scale-Adaptive Convolutional Neural Network". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2018), pp. 1113–1121.

[9] Xinya Chen et al. "Scale Pyramid Network for Crowd Counting". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2019), pp. 1941–1950.

[10] Q. Dang et al. "Deep learning based 2D human pose estimation: A survey". In: *Tsinghua Science and Technology* 24.6 (2019), pp. 663–676.

[11] Guangshuai Gao et al. "CNN-based Density Estimation and Crowd Counting: A Survey". In: *ArXiv* abs/2003.12783 (2020).