aleju / papers

<> Code Issues 4 Pull requests 1 Actions Projects Wiki Secur

master ▾ ···

papers / neural-nets / Stacked_Hourglass_Networks_for_Human_Pose_Estimation.md

aleju Fix typo History

1 contributor

Raw Blame

55 lines (48 sloc) 3.46 KB

Paper

- **Title**: Stacked Hourglass Networks for Human Pose Estimation
- **Authors**: Alejandro Newell, Kaiyu Yang, Jia Deng
- **Link**: https://arxiv.org/abs/1603.06937
- **Tags**: Neural Network, pose estimation
- **Year**: 2016

See also

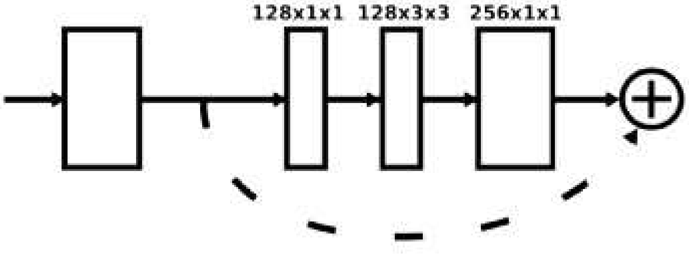
- Official code: https://github.com/anewell/pose-hg-train

Summary

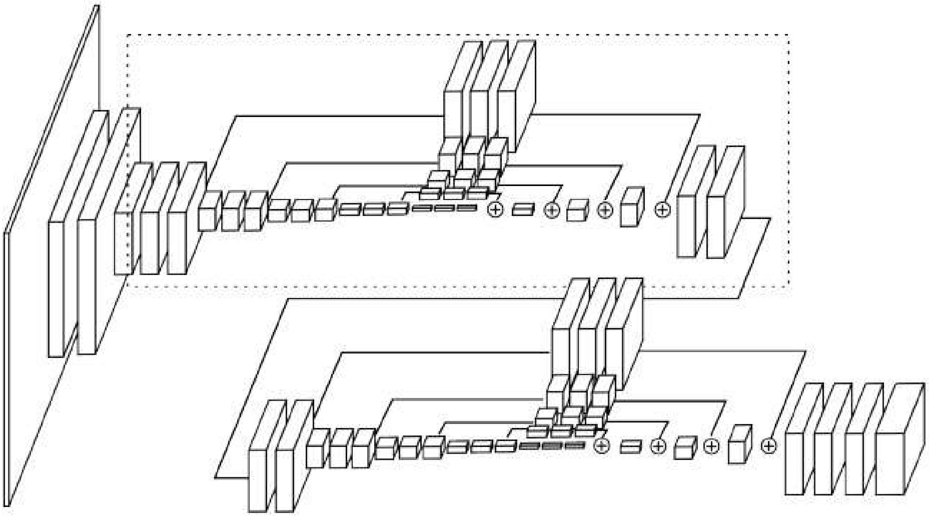
- What
  - They suggest a new model architecture for human pose estimation (i.e. "lay a skeleton over a person").

  - Their architecture is based progressive pooling followed by progressive upsampling, creating an hourglass form.
  - Input are images showing a person's body.
  - Outputs are K heatmaps (for K body joints), with each heatmap showing the likely position of a single joint on the person (e.g. "akle", "wrist", "left hand", ...).

- How

  - *Basic building block*
    - They use residuals as their basic building block.
    - Each residual has three layers: One 1x1 convolution for dimensionality reduction (from 256 to 128 channels), a 3x3 convolution, a 1x1 convolution for dimensionality increase (back to 256).
    - Visualized:
    - 

  - *Architecture*
    - Their architecture starts with one standard 7x7 convolutions that has strides of (2, 2).
    - They use MaxPooling (2x2, strides of (2, 2)) to downsample the images/feature maps.
    - They use Nearest Neighbour upsampling (factor 2) to upsample the images/feature maps.
    - After every pooling step they add three of their basic building blocks.
    - Before each pooling step they branch off the current feature map as a minor branch and apply three basic building blocks to it. Then they add it back to the main branch after that one has been upsampeled again to the original size.
    - The feature maps between each basic building block have (usually) 256 channels.
    - Their HourGlass ends in two 1x1 convolutions that create the heatmaps.
    - They stack two of their HourGlass networks after each other. Between them they place an intermediate loss. That way, the second network can learn to improve the predictions of the first network.
    - Architecture visualized:



- *Heatmaps*
  - The output generated by the network are heatmaps, one per joint.
  - Each ground truth heatmap has a small gaussian peak at the correct position of a joint, everything else has value 0.
  - If a joint isn't visible, the ground truth heatmap for that joint is all zeros.
- *Other stuff*
  - They use batch normalization.
  - Activation functions are ReLUs.
  - They use RMSprob as their optimizer.
  - Implemented in Torch.

- Results
  - They train and test on FLIC (only one HourGlass) and MPII (two stacked HourGlass networks).
  - Training is done with augmentations (horizontal flip, up to 30 degress rotation, scaling, no translation to keep the body of interest in the center of the image).
  - Evaluation is done via PCK@0.2 (i.e. percentage of predicted keypoints that are within 0.2 head sizes of their ground truth annotation (head size of the specific body)).
  - Results on FLIC are at >95%.
  - Results on MPII are between 80.6% (ankle) and 97.6% (head). Average is 89.4%.
  - Using two stacked HourGlass networks performs around 3% better than one HourGlass network (even when adjusting for parameters).
  - Training time was 5 days on a Titan X (9xx generation).