Machine Perception 2018 – Project Report

Hand Joint Recognition

Ivan Tishchenko
ETH Zürich
tivan@student.ethz.ch

Mickey Vänskä
ETH Zürich
mickeyv@student.ethz.ch

ABSTRACT

The advances of scientific methods in the area of artificial intelligence (e.g. computers becoming professional Go players) allow researchers to turn dreams into reality. A known challenge in the fields of computer vision and machine learning is the extraction of specific information from images and videos. One concrete application is detecting people and related information in images, as for instance human pose estimation or motion prediction to avoid collisions in driverless cars. These fields have seen increasing attention lately, however there are relatively few papers available on data extraction from specific body parts, such as hands. Hand pose estimation proves to be a complex challenge due to the high rate of self-occlusion and agile poses combined with the large number of joints concentrated in a small volume. This project explores the use of various CNNs for annotating respective joints in images of hands and concludes a modified Inception-ResNet-v2 to be the fittest. In direct comparison with a custom ResNet34, the network is expected to perform better by being a continuation of Residual Networks combined with Inception, thus extracting more features at each step. A CPM was also designed and showed a shallower accuracy slope at later epochs. Suprisingly, in the case of Residual Networks more layers led to worse performance. Even though Residual Networks and Inception-ResNet-v2 were designed for classification tasks, their use in regression tasks is still a viable solution. Our results demonstrate the viability of Inception-ResNet-v2 in the scope of joint estimation and we expect to see further work based on them in the field of pose estimation. Generally we expect to see more network designs in the future which allow generic feature extraction and are easily adaptable with high accuracy for many tasks. However, specialized networks will always exist for very high accuracy.

1 INTRODUCTION

In recent years advances in natural language understanding have led to a rise in voice-recognition systems from various companies with human-level performance in understanding commands and synthesizing human voices to respond.

The interaction in such cases is based solely on audio and as such not universally usable by those with hearing impairments or in noisy environments. An alternative would be using computer vision where communication occurs through sign language or by mapping hand poses to specific user-defined commands.

We begin by presenting relevant CNNs, their use in pose estimations and recent results in its domain. What follows is a description of our model we used to detect hand joints with a summary on results and accompanying discussion. Finally, we summarize our findings as well as suggest areas which are likely to lead to fitter models.

2 RELATED WORK

Current state of the art convolutional networks for image classification are Inception-v3 [10] and Inception-v4 with Inception-ResNet [9] developed by Szegedy et al. The latter network design is an amalgamation of the core principle of Inception-v1 and Residual Networks [4] invented by He et al. Residual Networks are the first successful approach to very deep convolutional neural networks for image classification which can easily scale to hundreds of layers and adapt to various classification tasks while being robust to vanishing gradients by heavy use of identity connections.

In human pose estimation work by Wei et al. is used heavily. The authors use a variant of VGGNet [8] combined with heatmaps to predict joint locations using Convolutional Pose Machines (CPM) [11]. CPMs were first designed to predict skeletal body poses, however adaptations have been made and included as part of pipelines to predict hand poses as in work by Zimmermann and Brox. The authors' approach supersedes our project's task by predicting 3D models of hands from a single RGB image [12] without the use of 3D depth data. The design of Convolutional Pose Machines is not restricted to using VGGNet as Insafutdinov et al. have shown with DeeperCut [6] by relying on a modified ResNet152.

Stacked Hourglass Networks [7] were investigated by Newell et al. to extract human pose estimation from single RGB images. Their network repeatedly downsamples and upsamples the input image in hourglass blocks and regresses using heatmaps joint locations. A combination based on Residual Networks and Stacked Hourglass Networks was envisioned by Bulat and Tzimiropoulos. Their design uses a part detector network followed by a part regression network.

3 METHOD

3.1 Problem Analysis

The task is, given an image representation $I \in \mathbb{R}^{3\times M\times N}$, to predict 21 joint locations of the hand $w_i = (x_i, y_i)$, where $i \in \{1 \dots 21\}$. Concretely in our case, 1 root point of the hand and 5×4 finger joints on RGB images of size 128×128 pixels where the hand is tightly cropped from source images of size 320×320 and scaled to the respective size.

The dataset [13] consists of 57'466 images of size 320×320 with accurately annotated joints for training as well as joint visibility information (excludes self-occlusion). A validation set consisting of 8'610 images is created from these 57'466 images. Predictions are generated for 3'611 tight crops of size 128×128.

3.2 Training Methodology

Networks are trained using L_2-loss on all joint predictions from ground truth labels. A secondary metric based only on visible joints

is also calculated, and for each L_2-loss accuracy scores are evaluated where each prediction in the vicinity of 2px (Euclidean distance) from the ground truth is marked as a positive result.

Batch sizes are set at 8, and AdamOptimizer is used with a fixed learning rate of 1e-4 at training time. Epochs are determined empirically on a per-network basis.

3.3 Model Selection

Multiple designs are explored based on work done by Wei et al. (Convolutional Pose Machines), Szegedy et al. (Inception-v3), He et al. (ResNet) & Szegedy et al. (Inception-ResNet-v2).

The evaluated Convolutional Pose Machine (CPM) uses 3 repetitive blocks as described in Figure 2 of [11] with 128 features extracted in each map and kernels of size 5×5. 22 heatmaps are predicted, 21 single joint locations and one with all joints subtracted to be used as background. To train the network each joint (x_i, y_i) of ground-truth labels is transformed into a heatmap of size 16×16 with a 2D-Gaussian with $\sigma = 1$ at the respective coordinate and one background map. The network is trained using an L_2-loss on predicted and ground-truth heatmaps using intermediate supervision to prevent vanishing gradients. Predictions are generated by upscaling predicted heatmaps bicubicly and extracting the maximum of each respective heatmap as an (x_i, y_i) coordinate.

The Residual Network architecture is a modified ResNet34 with average pooling removed and using `N × [Conv BatchNorm ReLU]` in basic building blocks (`N` being subject to the block type; for our approach `N` = 2). The last `ReLU` is discarded in building blocks as it shows to increase predictions in classification tasks as per [2]. Predictions are generated by flattening the output and adding one fully connected layer with 42 neurons before reformatting them to $(x_i, y_i) \quad i = \{1 \dots 21\}$. The network is trained using an L_2-loss on the predictions and ground-truth labels.

Using Inception-ResNet-v2 requires modifications to the stem and due to larger expected images (299×299) kernels of size 7 are scaled down to 5. The last MaxPool operation in Figure 3 of [9] is modified to not downscale by applying two convolutional blocks with 3×1 kernels. Activation scaling as shown in Figure 20 is set to 0.17 in Figure 16, 0.1 in Figure 17 and 0.2 in Figure 19 from [3]. Each convolution (unless linear) is followed by batch normalization and Leaky ReLU with $\alpha = 0.01$. Lastly the average pooling layer is set to not scale down but keep dimensions before flattening, using 42 output neurons and reformatting as with our ResNet34 architecture. Training and predicting are identical for both networks.

Inception-v3 is a stock implementation of the architecture from [10]. Global average pooling is used at the end. Training and prediction procedures are identical to ResNet34.

3.4 Data Augmentation

We experienced the tendency that the networks overfit during training time by converging with their value of the loss function towards 0, which is a known indication of overfitting. To solve this issue a randomization step is introduced during image feeding time to randomly sample from $X \sim Bern(p)$ on a sliding window of 4'000 images for the current batch. Additionally, each image is subject to being augmented with a probability $p_{aug} = \frac{3}{10}$ using the following ordering of operations: *rotate* [-90°, 90°], *shear* [-15°, 15°], **tight-crop**, **rescale**, *flip-horizontal*, *flip-vertical*, contrast [0.8, 1.2], brightness [0.8, 1.2], dropout and salt-and-pepper (each operation performed with probability $p_{op} = \frac{1}{2}$ on augmentation, operations in **bold** with $p_{op} = 1$ disregarding augmentation). Operations in *cursive* verify joints not escaping the image. Should this occur then the operation is rolled back.

Depth/Customization (Epochs)	Kaggle-Score
34/None (16)	Weak performance (aborted)
34/*lR, avg_same*	93.97
34/*lR, fPre, avg_same*	107.54
68/*DeeperCut*	103.43
34/*B, -avg*	108.99
34/*-avg*	89.28

Table 1: ResNet scores at 45 epochs, batch size 8 and similar data augmentation. Legend: *lR* = Leaky ReLU ($\alpha = 0.01$); *-avg* = removal of average pooling; *avg_same* = average pooling with `same` padding; *fPre* = full pre-activation blocks; *B* = bottleneck blocks; *DeeperCut* = shallower variant of [6] with our training/prediction approach.

Network (Epochs)	Public	Private
CPM (40)	86.37	85.09
Inception-v3 (40)	125.98	125.66
ResNet34 (55)	84.21	81.75
Inception-ResNet-v2 (108)	61.15	59.71

Table 2: Summary of final Kaggle scores for best network per architecture explored.

4 RESULTS

Table 1 show various variants of ResNet performing differently well at regressing hand poses using similar augmentation strategies (the last two do not apply salt-and-pepper & dropout and have $p_{aug} = \frac{1}{2}$). The last run is chosen for further comparisons.

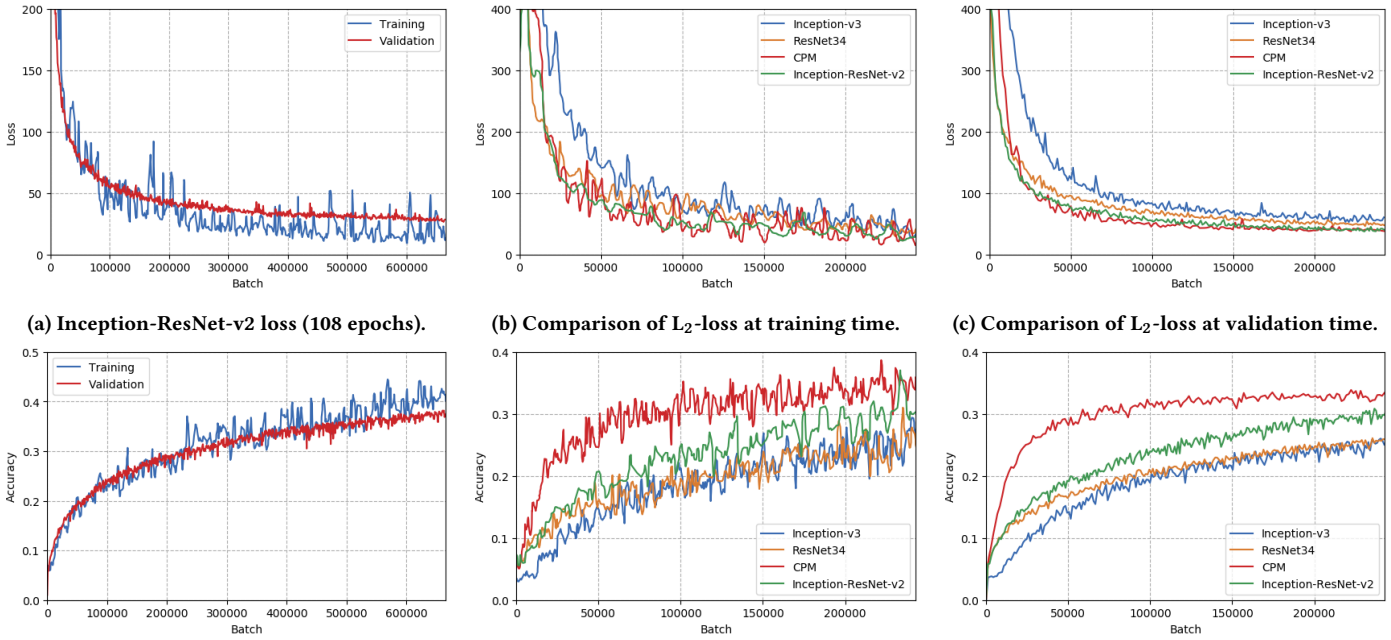
Table 2 summarizes the scores on Kaggle and training epochs for our network implementations.

Figure 1a shows the loss function of the best performing model (Inception-ResNet-v2) over 108 epochs. Its prediction accuracy can be seen in Figure 1d.

Training and validation losses of our four networks are presented in Figures 1b and 1c. Figures 1e and 1f give respective accuracy benchmarks. Graphs are clipped at 40 epochs to allow easy comparisons.

5 DISCUSSION

Considering the comparison of losses for our best model (Figure 1a) at training and validation time, it is clear that the model started to converge towards the optimal state which may have not been reached even after 108 epochs of training. Another indicator is the increasing prediction accuracy at training and testing time. Significant gains through additional training are not expected without giving the network new data. In comparison to other networks at 45 epochs it gave an accuracy with the largest slope while having



(a) Inception-ResNet-v2 loss (108 epochs). **(b) Comparison of L_2-loss at training time.** **(c) Comparison of L_2-loss at validation time.**

(d) Inception-ResNet-v2 accuracy (108 epochs). **(e) Comparison of accuracy at training time.** **(f) Comparison of accuracy at validation time.**

Figure 1: Score metrics of Inception-ResNet-v2 and direct comparisons of it with ResNet34, our CPM and Inception-v3.

the lowest loss. Therefore, it was decided to continue to train it for another 63 epochs.

Sufficiently good results were delivered by our ResNet34 as described earlier. The network's loss was the third smallest and is directly comparable to Inception-ResNet-v2 by using the same approach to loss calculations. The loss was consistently higher while the accuracy was overall lower. This network also trained quicker than Inception-ResNet-v2 but due to different GPUs used, no comparable numbers were produced in time.

Comparing different ResNet implementations it is interesting to observe the bottleneck layers which resulted in much worse predictions for the same network as well as advanced techniques presented by He et al. [5] detrimentally affecting network performances. The use of Leaky ReLUs is uncertain as of this point, further investigation is required to exclude side-effects of the last layer (Table 1) however there was not enough time for this. The last observation which is of interest and lacked time for thorough investigation as well is the shallower implementation of Insafutdinov et al. [6] to perform worse than our ResNet34.

Our CPM performed the best for low-epoch runs. Accuracy scores were much higher than any other network while the loss was the lowest. This behaviour is of debate to result in good scores for many epochs as accuracy and loss plateau significantly. Nevertheless, Kaggle scores evaluated at 20, 30 and 40 epochs (94.89, 97.51, and 86.37) showed increases in performance. Modifications to our CPM (increasing the depth, using larger heatmaps) are expected to increase the fitness.

Inception-v3 exhibited the worst evaluation metrics. The loss decreased less rapidly and finished with overall higher values. A similar accuracy as ResNet34, but higher loss confirmed the received Kaggle score.

Of interesting note was that validation losses did not map with the Kaggle score. The generated validation set and the test images are therefore from different distributions.

The scores received on Kaggle (Table 2) show minor differences except for ResNet34. Overall, model predictions are robust and exhibit predictable performance for future tasks.

6 CONCLUSION

In our work we have shown the use of a modified Inception-ResNet-v2 to be a viable alternative to Convolutional Pose Machines in detecting hand joints in images. The performance and fitness of the model show promising results and make it an interesting candidate for future networks on the task of 2D hand pose estimation from single RGB images. With the network being a complex architecture we would like to note its low training throughput in comparison to the other three designs. An interesting approach would be to take the work designed by Bulat and Tzimiropoulos and have Inception-ResNet-v2 as the part regression network. More involved approaches would include using a model of a hand which is morphed into the prediction of the network and verified to be a reachable pose according to human anatomy. Using this network on videos of hands instead of single RGB images would need further modifications to get smooth predictions with the help of a recurrent network. As of now this network would likely result in high scores for single RGB image poses but sequences of hand-motions are still to be researched. Finally, using our network as the 2D detector of Zimmermann and Brox [12] could boost the generated 3D models significantly.

REFERENCES

[1] Adrian Bulat and Georgios Tzimiropoulos. 2016. Human pose estimation via Convolutional Part Heatmap Regression. *CoRR* abs/1609.01743 (2016). arXiv:1609.01743 http://arxiv.org/abs/1609.01743

[2] Sam Gross and Michael Wilber. 2016. Training and investigating Residual Nets. http://torch.ch/blog/2016/02/04/resnets.html

[3] Sergio Guadarrama and Neal Wu. 2017. Tensorflow. https://github.com/tensorflow.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity Mappings in Deep Residual Networks. *CoRR* abs/1603.05027 (2016). arXiv:1603.05027 http://arxiv.org/abs/1603.05027

[6] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. 2016. DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. *CoRR* abs/1605.03170 (2016). arXiv:1605.03170 http://arxiv.org/abs/1605.03170

[7] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. *CoRR* abs/1603.06937 (2016). arXiv:1603.06937 http://arxiv.org/abs/1603.06937

[8] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[9] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *CoRR* abs/1602.07261 (2016). arXiv:1602.07261 http://arxiv.org/abs/1602.07261

[10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. *CoRR* abs/1512.00567 (2015). arXiv:1512.00567 http://arxiv.org/abs/1512.00567

[11] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. *CoRR* abs/1602.00134 (2016). arXiv:1602.00134 http://arxiv.org/abs/1602.00134

[12] Christian Zimmermann and Thomas Brox. 2017. Learning to Estimate 3D Hand Pose from Single RGB Images. *CoRR* abs/1705.01389 (2017). arXiv:1705.01389 http://arxiv.org/abs/1705.01389

[13] Christian Zimmermann and Thomas Brox. 2017. *Learning to Estimate 3D Hand Pose from Single RGB Images.* Technical Report. arXiv:1705.01389. https://lmb.informatik.uni-freiburg.de/projects/hand3d/ https://arxiv.org/abs/1705.01389.