Hand Joint Recognition

Paul-Edouard Sarlin
psarlin@student.ethz.ch

Moritz Zimmermann
zimoritz@student.ethz.ch

1 INTRODUCTION

Human interaction is a manifold mixture of verbal and non-verbal communication with speech and gestures being just two out of many of the involved tools. To the contrary human-machine interaction is significantly limited. Despite recent advances in speech recognition it is still mostly limited to physical input devices. The advent of deep learning has greatly improved machine perception in many ways, specifically in regards to visual recognition and localization. In this project we implemented a covolutional neural network that predicts joint positions of a human hand from images. Reliably localizing pose and motion of all hand joints could be a valuable element in many applications such as gesture control or tele-manipulation.

In this report, we present our experiments on the RHD dataset [9], composed of synthetic images of human hands in various poses and backgrounds, with accurate ground truth of the hand joint locations. Taking inspiration from state-of-the-art models in pose estimation, our approach achieves a competitive score of 47.97 on the public Kaggle leaderboard, granting us the 2nd position with 3 times fewer submissions than other competitors who reach the hard baseline.

2 RELATED WORK

Recent advances in GPU hardware acceleration and large scale image datasets [2] have enabled deep learning to significantly improve state-of-the-art results in many computer vision tasks. Recent hand joint recognition methods are thus mainly based on convolutional neural networks (CNN).

Related to the problem that we tackle is the work of Zimmermann and Brox [9], who proposed to predict a 3D hand pose in a three-step fashion. Their method first segments a mask of the hand, then regresses a Gaussian score map for each joint individually, and finally refines these predictions into accurate 3D keypoint locations. This architecture is, to some extent, also applicable to the prediction of 2D poses.

The problem of human pose estimation is very similar to hand joint recognition and has recently seen significantly more progress. As such, Toshev and Szegedy [7] showed impressive results by directly regressing the 2D Cartesian joint locations, while Wei et al. [8] proposed a fully-convolutional network to learn translation-covariant scores. Both methods were based on multi-stage networks that iteratively refined the predictions.

Such complex networks have been made unnecessary by recent advances in powerful feature encoders, such as VGG [6] or ResNet [3], whose high number of layers allows to learn useful high-level representations. As such, current state-of-the-art human pose estimation methods [4, 5] leverage these high-quality features, along with large labeled datasets, to train very deep encoder-decoder architectures. Using similar techniques as for CNN-based semantic segmentation [1], a ResNet encoder first computes a high-level feature map with low spatial resolution, which is subsequently bilinearly upsampled. It is then fed into two network heads, which, for each joint, first predict a coarse classification of the joint location, and secondly refine it by prediciting 2D offsets.

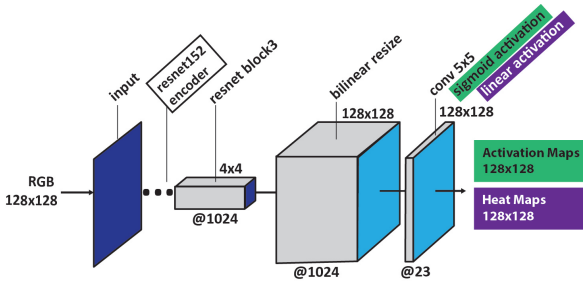

Figure 1: *Fully convolutional head.* We bilinearily upsample the feature map to image resolution and apply a 5x5 convolution to restore spatial information. The final activation function is sigmoidal for pixel-wise classification (green) and linear for score map regression (purple).

We take inspiration from these diverse methods, but adapt them to our problem and update them, when relevant, with recent state-of-the-art architectures.

3 METHOD

The given problem differs from the ones usually addressed by past works in several manner. As the hands are already cropped, no initial hand segmentation step is required. Additionally, the input image have a fairly low spatial resolution (128×128), which makes it difficult to blindy apply encoder-decoder architectures. In this section, we first justify our choice of decoder, and subsequently detail the three decoders that we experiment with.

3.1 Encoder Architecture

Following its use in many state-of-the-art architectures across a wide range of vision tasks, we use a ResNet [3] encoder to extract high-level features from the input image. We more specifically employ ResNet-152 up to its third block, comprising a total of 142 layers. This restricted ResNet allows to preserve some spatial information necessary for the decoder. The significant depth of this design comes at the benefit of a receptive field that is a multiple of the input image size, allowing to encode image-wide high-level features, such as the type of hand pose.

As the dataset is fairly small and repetitive, training the encoder from scratch presents a high risk of overfitting. We thus initialize it with weights trained for ImageNet classification [2], which have been shown to generalize well to other tasks.

3.2 Decoder Architecture

We evaluate three classical approaches to decode the high-level feature map into accurate joint locations.

3.2.1 Pixel-wise classification. We first cast joint recognition as a pixel-wise multi-label binary classification. We aim at predicting for each pixel if it contains one of the 21 keypoints. The label is sparse, as only a single pixel per image and per keypoint is 1, making the classification extremely unbalanced and the learning very challenging.

Inspired by the work of Papandreou et al. [5], we perform a binary classification to predict whether a given pixel is within a distance r of a keypoint P_i or not. In essence, this is equivalent to predicting if a given pixel is member of a disk of radius r_i centered at the joint location. This new coarse labelling leads to more balanced classes, making the training tractable. Some visual examples are shown in Figure 2.

We solve this classification problem for all 21 keypoints at once by generating one activation map per keypoint. We predict activation maps in a fully convolutional fashion with a layout as depicted in Figure 1. Given the resulting keypoints probability map $p(x, y)$, we extract 2D Cartesian coordinates using the expectation value to achieve sub-pixel accuracy:

$$\hat{x}, \hat{y} = \mathbb{E}_{x,y}[p(x, y)]$$

3.2.2 Heat map regression. Our second decoding approach directly regresses softly localized score maps [9]. This idea originates from the same intent to tackle sparsity and convergence issues that come with the high localization of the problem. The target score maps are generated as 2D normal distributions centered around the keypoint locations and with variance σ^2. The model is trained by minimizing the L2 distance between the predicted and the target score maps. As previously, 2D Cartesian coordinates are computed as the expectation over the normalized location scores.

3.2.3 Direct regression. The third decoder that we considered is a direct regression of 2D Cartesian coordinates in a fully connected fashion. As opposed to the previous fully-convolutional decoders, this approach is not translation-invariant, and is thus not robust to differences in the keypoint location distributions of the training and test sets. We nonetheless consider this as a serious baseline, as our test and training sets come from the exact same synthetic data distribution. As it does not exploit any spatial information, we use the feature map from the 4th block of ResNet (size $1 \times 1 \times 1024$) and apply a fully-connected layer with 2×21 units and linear activations.

3.3 Data Augmentation

Augmenting training images accounts for a significant part of our success on the Kaggle leaderboard. We apply on-the-fly random distortions to the training images in the form of brightness and saturation changes as well as rotations, such that no training example is seen twice by the network.

We found test-time augmentation to be also largely beneficial. For each image of the test set, we predict keypoints on its four rotated variants and compute their average.

4 EXPERIMENTAL RESULTS

We first provide some implementation details. All networks are trained with a batch size of 32 using RMSProp until convergence on a validation set of 480 samples (approximately 120k iterations). Regarding the extended labels of the fully-convolutional methods, there is clear trade-off between the sparsity of the labels and their spatial localization, resulting in a trade-off between the ease of training and the spatial accuracy of the predictions. We found that a disk radius r of 16 and a score map variance σ^2 of 100 provide a reasonable balance.

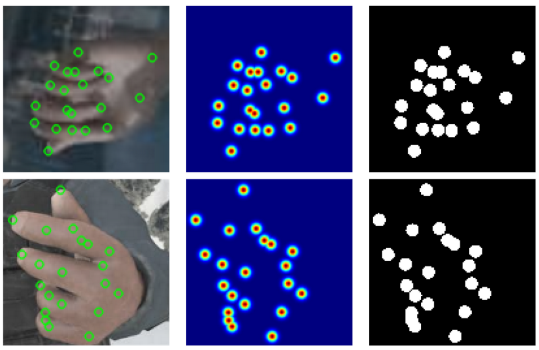

Figure 2: *Our relabelling of the dataset.* We show the input image (left) with keypoints drawn in green, the score maps (center) and the binary disks (right). Their respective variance and radius are in practice larger to counter sparsity.

Model	MSE
Fully Connected	47.97
Pixel-wise classification	69.99
Scoremap regression	443.42

Table 1: Mean Squarred Error (MSE) of the different architetures on the Kaggle public leaderboard.

Model	MSE
FC+centered-crop	189.99
FC+biased-crop	139.07
FC+biased-crop+train-aug	62.38
FC+biased-crop+train-aug+test-aug	47.97

Table 2: Influence of the data pre-processing on our fully-connected (FC) model.

We summarize the accuracy of our models in Table 1. Surprisingly, the fully-connected head outperforms the other models by a large margin. The score map head performs poorly, which is not unexpected as dense regression is notably more difficult than dense binary classification. Visual samples are shown in Figures 3 and 4.

We also perform an ablation study on the fully-connected (FC) head to analyse the influence of the data pre-processing and show the results in Table 2. The original skeleton code crops the hands from the full original images in a way that biases the keypoint towards the upper-left corner of the patch. Statistics on the training set show that this results in an average keypoint location of $(60, 54)$ instead of $(64, 64)$ (patch center). We correct this bias (FC+centered-crop) and compare it to the original cropping method (FC+biased-crop). We further evaluate the contribution of performing data augmentation on the training set (train-aug) as well as at test time (test-aug). The results show that correcting the bias deteriorates the

accuracy. We explain this as follows: the test set is generated using this same biased cropping method. Correcting it in the training set only creates a significant difference between the training and test distributions, which the FC model is very sensitive to. We expect fully-convolutional models, which are translation-covariant, to be robust to this kind of difference. On the other hand, data augmentation helps to increase the variance of both training and test distributions, such that their overlap is artificially increased.

Figure 3: *Visual samples of predictions of the classification head.* We show the input image (left) with its disk label of radius 20 (blue) and the predicted probability map (right), where the expectation values are marked in yellow.

Figure 4: *Visual samples of predictions of the fully-connected head.* The predicted keypoints are shown in green.

5 DISCUSSION

Our ablation study highlights the bias of the dataset as well as the weakness of the FC model. It however still significantly outperforms our other models despite their state-of-the-art results in human pose estimation. This can be due to the nature of the dataset that we work with, which is fairly simple compared to real datasets tackled in the literature. As such, the low input resolution yields very small encoded feature maps with spatial resolution 4×4, which encode much more information in their channels than in their spatial location. Despite an up-sampling step, this impairs the benefit of applying further convolutions and makes it easy for a fully-connected head to learn the hand poses and directly regress the joint locations.

Such a simple model would not generalize well to a real-world dataset, where accurately cropping the hands is difficult, as our model shows to be very sensitive to this pre-processing step. On the other hand, fully-convolutional models should be robust to it, but are not able to express their full potential on such a dataset.

It is also worth noting that other criteria, such as model size or run-time, might be important in real applications. These parameters notably do not matters in Kaggle competitions, allowing us to use one of the largest models available, and apply an expensive test-time augmentation that slows down the inference by four times but leads to a significant boost in accuracy. In a real use case, accuracy might not be the paramount priority.

6 CONCLUSION

In this work, we implemented three different models that solve the problem of hand joint recognition. These are based on a common encoder but differ in their decoders, which perform pixel-wise classification, dense score regression, and direct regression of 2D Cartesian coordinates. Two of them score significantly higher than the hard baseline and, despite our expectations, the direct regression model grant us the 2nd place on the public Kaggle leaderboard. Heavy data augmentation is also a major part of our performance.

The success of the fully-connected model is however fragile, and we highlighted some of its limitations. Further experiments would need to validate that fully-convolutional models are superior when translation-invariance is necessary, e.g. when the input cannot be perfectly cropped, or when input images might have various sizes. To achieve higher accuracy, a subsequent refinement of the predicted locations might be necessary, for example using a second head that predicts refinement offsets and performs a Hough voting scheme over them [5].

REFERENCES

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611* (2018).

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 248–255.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[4] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. 2018. PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model. *arXiv preprint arXiv:1803.08225* (2018).

[5] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. 2017. Towards accurate multiperson pose estimation in the wild. *arXiv preprint arXiv:1701.01779* 8 (2017).

[6] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[7] Alexander Toshev and Christian Szegedy. 2014. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1653–1660.

[8] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4724–4732.

[9] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3d hand pose from single rgb images. In *International Conference on Computer Vision*.