Human Pose Estimation and Joint Detection: Semester Project Report

Can Yilmaz Altinigne
Department of Computer Science
École polytechnique fédérale de Lausanne
can.altinigne@epfl.ch

***Abstract*—In this project, we aim to train a fully convolutional network for human pose segmentation. For this purpose, we use U-Net architecture which is mainly used in segmentation tasks. Additionally, we change the convolution layers with the harmonic convolution layers to test if we can make learning faster. Then we train a combined model based on the best segmentation model to find accurate human pose segmentation masks and joint locations on human body concurrently.**

***Index Terms*—image segmentation, joint location prediction, fully convolutional networks, harmonic networks**

I. Introduction

Since the use of convolutional neural networks in different computer vision tasks increases, some network models present remarkable results in human segmentation and human pose estimation tasks. A model called Mask-RCNN can provide the detection and creation of accurate segmentation masks of an object concurrently [1]. Moreover, some models such as OpenPose can detect locations of human joints and draw limbs between them in a real-time multi-person image and a video. The restoration of human joints and limbs in 3D can also be achieved by OpenPose [2].

It is a known fact that the performance of deep neural networks rises with the number of data points in the dataset. However, for a lot of tasks, it is not trivial to gather enough data to obtain a high-performance model. Some different model architectures such as U-Net architecture is developed in order to solve this problem [3]. U-Net is a fully-convolutional network which means that it does not use any feed-forward layer, it only uses convolution layers. U-Net architecture produces promising results in biomedical image segmentation tasks with the small number of training images. U-Net achieved state of the art results for several segmentation competitions.

In addition to the problem of less number of training images, another problem of convolutional neural networks is learning rotation invariance. This problem is generally solved by data augmentation methods. Harmonic networks aim to solve this problem by converting conventional convolutional layers to circular harmonic filters [4].

In this project, it is aimed to use state of the art models in human pose estimation and joint detection, and obtain promising results for both tasks with a combined model. For this purpose, we create different datasets which include single-person full body images in order to train our models. We use OpenPose for creating target values for joint locations, and Mask-RCNN in order to generate target human pose masks.

Then we compare the performance of regular convolutions and harmonic convolutions in U-Net architecture. Moreover, we build a combined model which can estimates human pose segmentation masks and joint locations with limbs in a single-person image simultaneously. We believe that this combined model would be useful in other research problems such as human weight and height estimation.

II. Materials and Methods

A. Training Set Preparation

We have created two different datasets for the training process. Both datasets are composed of three different datasets which have single-person full body human images. The reason why we choose single-person images is that we do not aim to do semantic segmentation. Data samples that we use in order to create our own datasets are from Leeds Sports Images Dataset [5], MPII Human Pose Dataset [6] and Fashion Dataset [7].

We have created a big dataset which has 18.6k images. We have only picked single-person images showing full-body of a person. OpenPose can determine the locations of 25 different joints in a human body [2]. In order to select full-body images, we eliminated the images that have the number of joints below 22.

However, training process was quite slow with the big dataset. Thus we have created another dataset composed of 5k images. Also, we rescaled smaller edges of all images to 128 pixels with Bicubic interpolation in order to expedite training process. The wider edge was also rescaled proportionately. We have trained all network models with the small dataset.

Fig. 1. Example images from the small dataset

B. Target Value Preparation

Human pose segmentation masks are created using Mask-RCNN. We considered Mask-RCNN outputs as target values. Also, we used OpenPose in order to save joint locations as (x,y) positions in a JSON file. Later, we used these JSON files to create Gaussian feature maps which we use during joint location prediction.

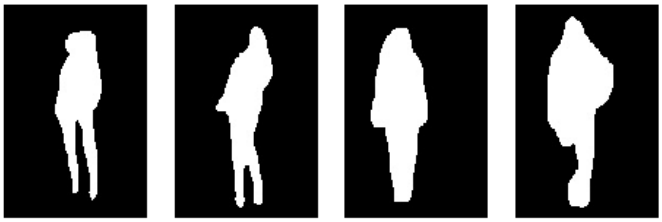

Fig. 2. Examples from created target values

C. Used Network Architectures

1) Segmentation with U-Net: U-Net is a fully convolutional network which is mainly used in biomedical image segmentation and it can present precise outputs with a small amount of training samples [3].

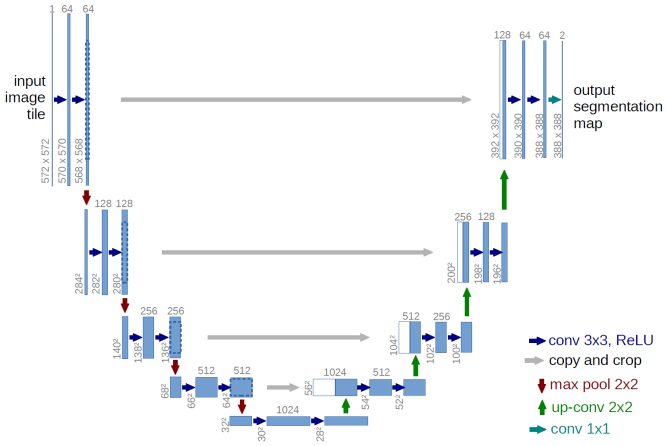

Fig. 3. Original U-Net Architecture [3]

It uses convolutions and pooling layers to decrease the image size, and then it uses either up-sampling layers or transposed convolutions to make the image size same again. During up-sampling, the network also concatenates the down-sampling outputs with the up-sampling outputs [3].

The architecture that we use has same number of up and down convolutions (4 layers), and it also uses the same number of channels (64 to 1024) in convolution layers. We use up-sampling instead of transposed convolutions. Moreover, we added dropout layers between up and down convolutions with the rate equal to 0.3.

The output layer is a pixelwise softmax layer with two channels. One channel is for background pixels and the other is for foreground pixels. The target value for background pixels is a binary image which has 1's in the pixels that is a background pixel in the original mask, and the same is also valid for foreground target values. We implemented U-Net models using Keras framework.

2) Segmentation with Harmonic Networks: We use the same structure as in U-Net model. However, we change convolution layers with harmonic convolutions using a library on GitHub [8]. We use harmonic networks in order to check if the learning process is faster than regular convolutions, and also we may utilize rotational invariance feature of harmonic networks even though the most images do not suffer from the rotation problem.

Harmonic netwoks use complex values in filters in order to solve rotation problem. Implementation of harmonic networks is based on concepts of rotation orders and streams [4].

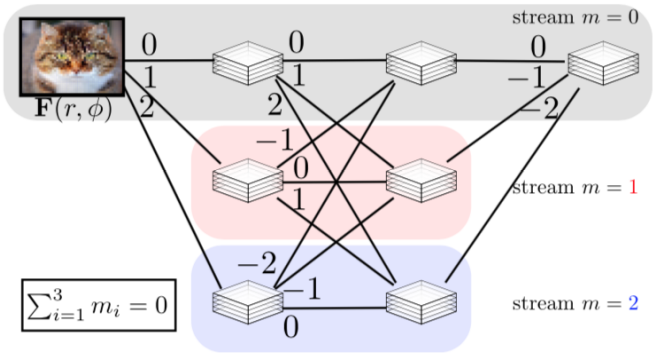

Fig. 4. H-Net Architecture [4]

Each stream in the figure above uses different angular frequency to process input image. There are three order (0th, 1st and 2nd) feature maps. Information is shared between layers. This architecture provides feature preservation during rotation. Since the optimal number of feature maps is still a research question, and the authors suggest only to use zeroth and first order feature maps, we only used 2 order feature maps.

Total number of order features was set as the number of neurons in U-Net architecture that we used. For example, if we use 64 neurons in regular convolutions, we use 32 0-order feature maps and 32 1-order feature maps.

D. Experiments

First, we compared the performance of U-Net and Harmonic Network on estimating human poses (masks). Then we proceeded with the best model on human pose estimation and joint detection simultaneously. We used the small dataset for all experiments. We set validation set size proportion as 0.2. We hereby have 4000 images for training and 1000 images for validation. We set the batch size as 1, since all images do not have the same size. However, using batch size of 1 was a bottleneck during training. We trained all models on a NVIDIA Tesla P100 GPU.

1) Training U-Net for only Human Pose Estimation: As indicated, we used the original U-Net architecture with a 2-channel softmax output to predict human poses. We used ADAM optimizer [9] with the learning rate equal to $2 \cdot 10^{-5}$. For mask prediction models, we used Dice loss as the loss function [10].

$$DiceLoss = 1 - \frac{2 \cdot |X \cap Y|}{|X| + |Y|} \quad (1)$$

We trained the model for 100 epochs, and saved the model weights which have the lowest validation loss. Loss function is named as Soft Dice Loss sometimes, because we only checked the softmax outputs (real values between 0-1) in the foreground channel and the target values in binary image masks.

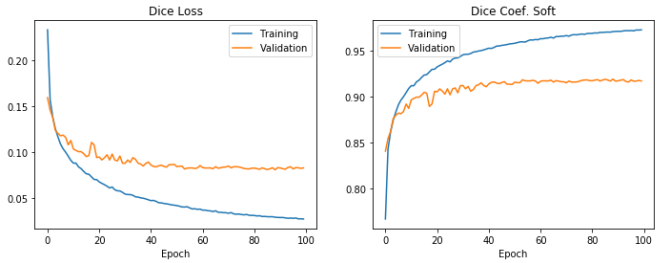

Fig. 5. Training U-Net for only Human Pose Segmentation

The lowest validation loss was in the 89th epoch. Training dice coeffcient was **0.9711** and validation dice coefficient was **0.9192**. Training process took 6.5 hours.

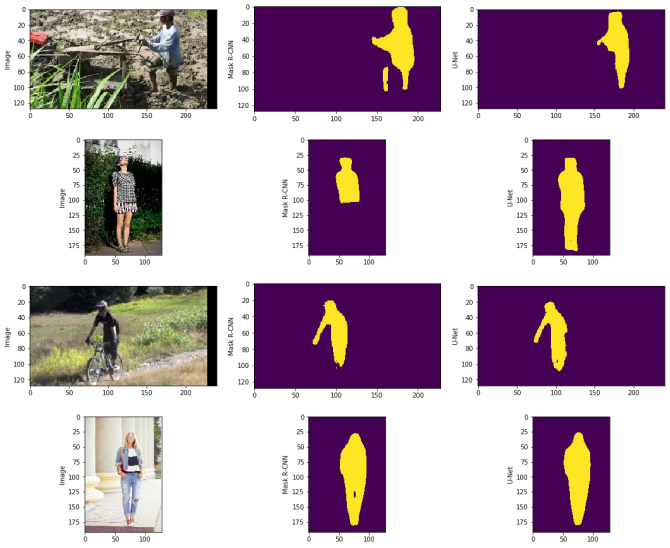

Fig. 6. Example validation set predictions of U-Net (the rightmost column)

2) Training Harmonic Network for only Human Pose Estimation: We kept the same structure as in U-Net implementation. Only the regular convolution layers are changed to harmonic convolutions. Other specifications for training process is the same as in U-Net experiment.

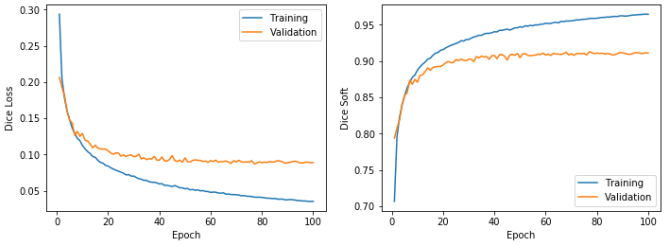

Fig. 7. Training H-Net for only Human Pose Segmentation

TABLE I
HUMAN POSE ESTIMATION: U-NET VS. HARMONIC NETWORK

	U-Net	**Harmonic Net**
Train. Dice Coef.	0.9711	0.964
Val. Dice Coef.	0.9192	0.911
Training Time	6.5 hours	30 hours

Harmonic network was implemented using Pytorch. However, since we set batch size as 1, each epoch took substantial amount of time compared to U-Net code which is written using Keras. Moreover, U-Net had better validation loss than Harmonic network.

However, one should not lead to a conclusion that having a higher dice coefficient is a better model for this task. Because, having a high dice coefficient means that the results are closer to Mask-RCNN outputs. On the other hand, as observed, the model can find more accurate human poses than Mask-RCNN does for some samples, and it causes a lower dice coefficient in the results. Therefore, we continue with U-Net architecture in next experiments mostly because of less training time.

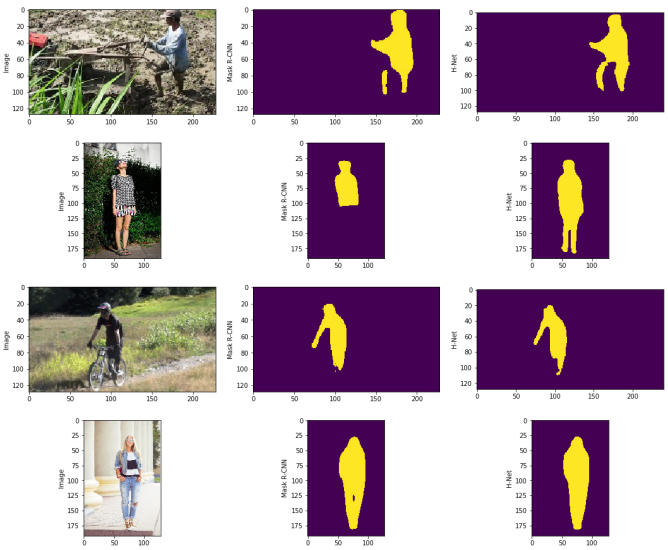

Fig. 8. Example validation set predictions of H-Net (the rightmost column)

3) Simultaneous Human Pose Estimation and Joint Detection: We predicted human pose masks and joint locations concurrently with U-Net model. For joint locations, we used different target values.

In order to create target values for joint locations, first we get the coordinates of joint centers from OpenPose outputs. Then we created a matrix with ones in corresponding joint center pixels and zeros in other pixels. After, we applied a gaussian filter on this matrix with sigma equal to 1. In the end, we normalized the outputs between 0 and 1. We call joint prediction target values as heatmaps.

a) Mask and Regression Heatmap Model: We trained a combined model which outputs human pose masks and predicts the heatmaps given in the figure below. We used Mean Squared Error for the joint location prediction. However, since values are between 0-1, the model could not learn well, and all pixels converge to zero. Then we renormalized the values between 0-100 in order to make model learn the joint locations. For mask prediction, the combined model have the same structure as in the previous experiments, it uses pixelwise softmax for foreground and background pixels.

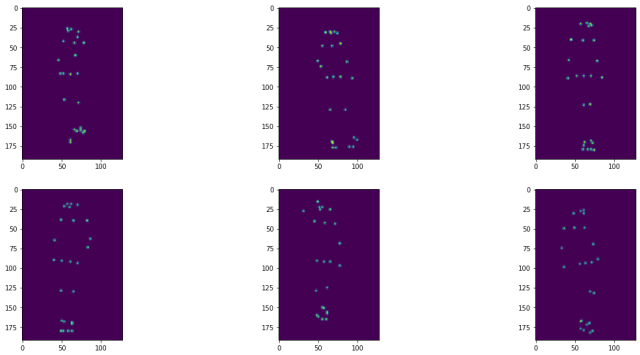

Fig. 9. Example heatmaps (values between 0 and 1)

For joint prediction, a convolution layer with linear activation function is used to predict the real values between 0-100. As a result, this approach changes joint location problem to a regression problem.

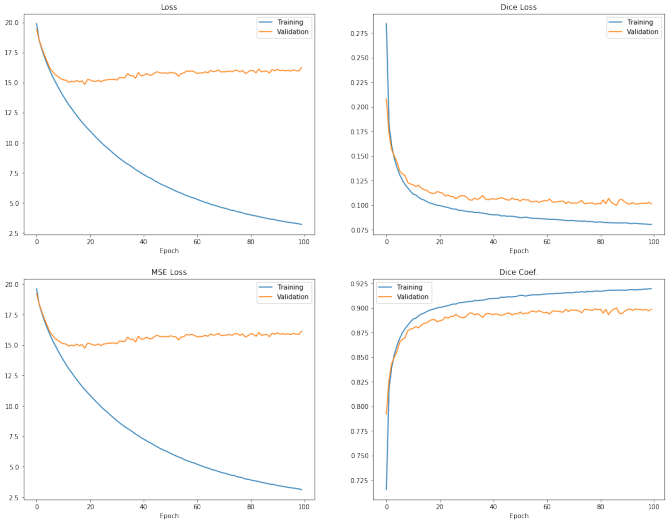

Fig. 10. Training a combined model for mask and joint location prediction (Real values between 0-100 in a 1-channel image)

We used ADAM optimizer with learning rate of $2 \cdot 10^{-5}$, and we trained the model for 100 epochs. Total error is the sum of MSE Loss for joints and Dice Loss for masks. Training dice coefficient is **0.899** and validation dice coefficient is **0.888**. Training MSE is **11.35** and validation MSE is **14.74**.

This approach for joint prediction seems working good. The results of this combined model can be seen in Figure 11. However, dice coefficient for mask predictions decreases by 0.2 with respect to the model that we predict masks only. Another problem of this model is that we do not know the specific joint locations, we only know that there is a joint with a certain possibility in those pixels.

Fig. 11. Combined Model Predictions (third and fifth columns)

In order to determine which pixel location corresponds to which joint, we converted target values to a 26-channel output using 25 channels for 25 joints, and 1 channel for background pixels. We performed two experiments using this approach.

b) Mask and Softmax Heatmap Model: After applying a gaussian filter on a binary image which has ones in the joint center pixels, we have different gaussians for each joint as in the Figure 9. Then we find the pixels that have a value higher than zero in this gaussian heatmaps. We find the closest joint center to these pixels, and change the pixel value with the number of the closest joint. At first, we use the pixels that have value more than zero in the heatmaps, however that led to squares for joints, then we changed the threshold to 0.01. This approach presents gaussian-looking joint locations. An example of this target value can be seen in Figure 12.

Target values have different values from 0 to 25, 1-25 for joints and 0 for background pixels. During training we created 26-channel output, and made the pixels 1 in the corresponding channels. For example, we set the pixels which correspond to the third joint in 2D heatmap (Fig. 12) to 1 in the third channel of the output. We used channel-wise softmax to predict which

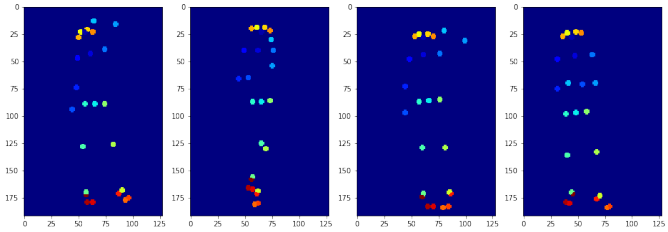

Fig. 12. Created joint prediction target values (each color represents different joint)

joint a pixel corresponds to. As the loss function, we used categorical cross entropy.

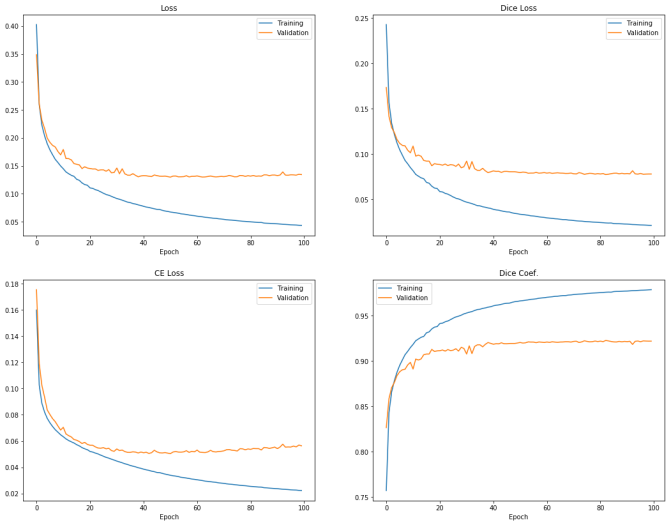

Fig. 13. Training process for combined mask and joint prediction (1's in channels)

Again, we used the same settings for the optimizer, and trained the model for 100 epochs. Total error is the sum of Cross Entropy for joints and Dice Loss for masks. Training dice coefficient is **0.978** and validation dice coefficient is **0.922**. Training cross entropy is **0.022** and validation cross entropy is **0.056**.

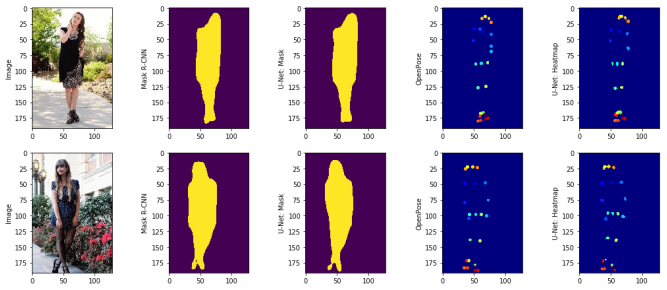

Fig. 14. Combined Mask and Joint Prediction with Softmax Class Outputs (the third and fifth columns)

With this approach, we are able to see the locations of specific joints as opposed to the previous experiment. Also, we observed that we have a higher dice coefficient for masks when we combined human pose estimation with the joint prediction.

c) Mask and Regression Softmax Heatmap Model: We made a small change to the previous approach in this experiment. Instead of having ones in joint pixels in corresponding channels, we set values of those pixels to gaussian outputs that we created and normalized between 0 and 1 during heatmap creation.

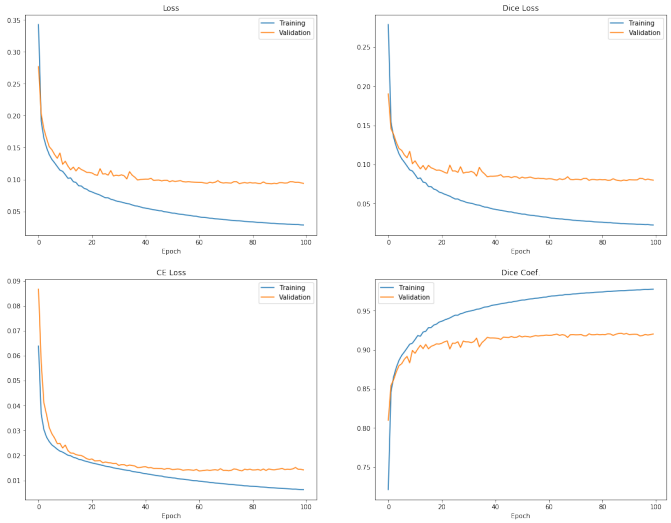

Fig. 15. Training process for combined mask and joint prediction (Real values between 0-1 in channels)

We again used categorical cross entropy, but this time with real values instead of ones. This approach can be considered as having joint confidence scores in pixels instead of if the existence of a joint in a pixel. Training process used the same settings as before, and the total error is the same as in the previous example. Training dice coefficient is **0.975** and validation dice coefficient is **0.921**. Training cross entropy is **0.007** and validation cross entropy is **0.014**.

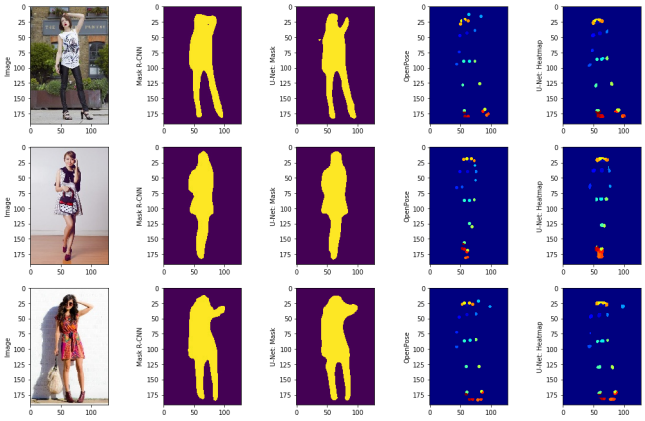

Fig. 16. Combined Mask and Joint Prediction with Softmax Regression Outputs (the third and fifth columns)

d) Triple Model: In this experiment, we combined the approach that we used in the second experiment with the regression approach in the first experiment for joint detection. We used three different target values: human pose masks, real values between 0 and 100 in pixels (the first experiment), and the ones in joint pixels in corresponding channels (the second experiment).

We used MSE for regression values and categorical cross entropy for softmax joint prediction. Again, we used Dice Loss for mask prediction. Total loss is the sum these losses. Training process is performed with the same settings.

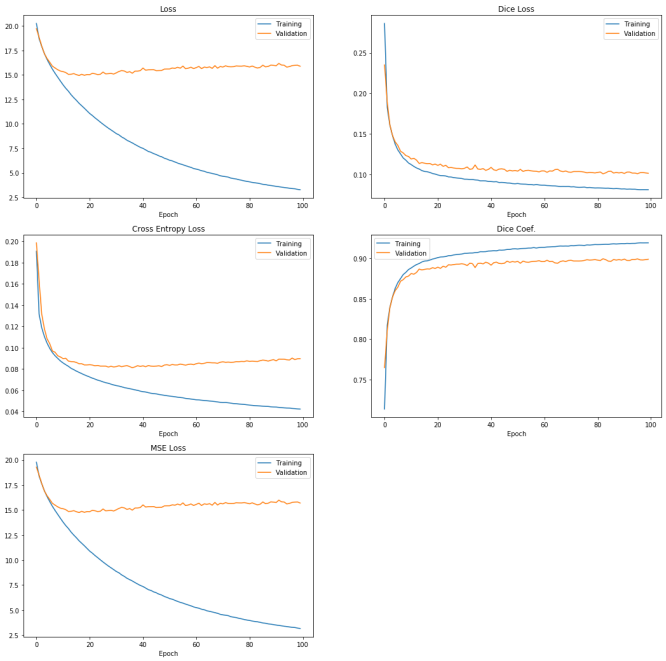

Fig. 17. Training process for combined mask and joint prediction using additional regression outputs

We used triple output model in order to check if the predicted joints are more accurate with the prediction of regression values.

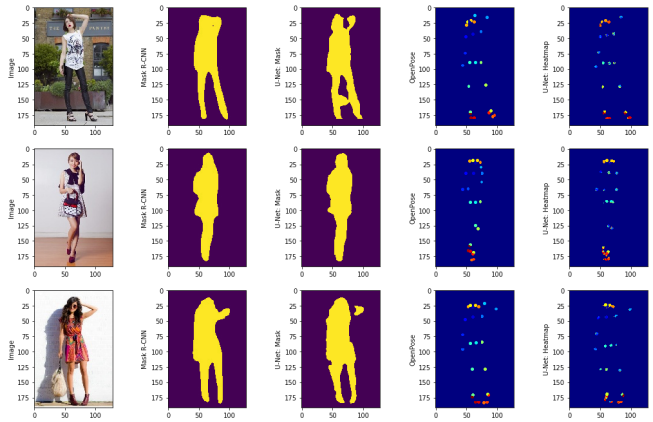

Fig. 18. Combined Mask and Softmax Joint Prediction using additional regression outputs (the third and fifth columns)

However, dice coefficient for mask predictions decreased by 0.03 using this approach, and there was no significant change in joint prediction. Training dice coefficient is **0.896** and validation dice coefficient is **0.886**. Training cross entropy is **0.076** and validation cross entropy is **0.084**.

III. Results

We performed four different experiments using U-Net architecture. We decided using U-Net over Harmonic Network because of its higher performance during human pose estimation and training duration. The results of four experiments are given below.

TABLE II
Model Performances

	Mask + Reg.	**Mask + Softmax**	**Mask + Reg. Softmax**	**Triple Model**
Training Dice Coef.	0.899	**0.978**	0.975	0.896
Training MSE	**11.35**	-	-	11.92
Training Cross Ent.	-	0.022	**0.007**	0.076
Validation Dice Coef.	0.888	**0.922**	0.921	0.886
Validation MSE	**14.74**	-	-	14.74
Validation Cross Ent.	-	0.056	**0.014**	0.084

The performance of Mask+Softmax and Mask+Regression Softmax models are very close.

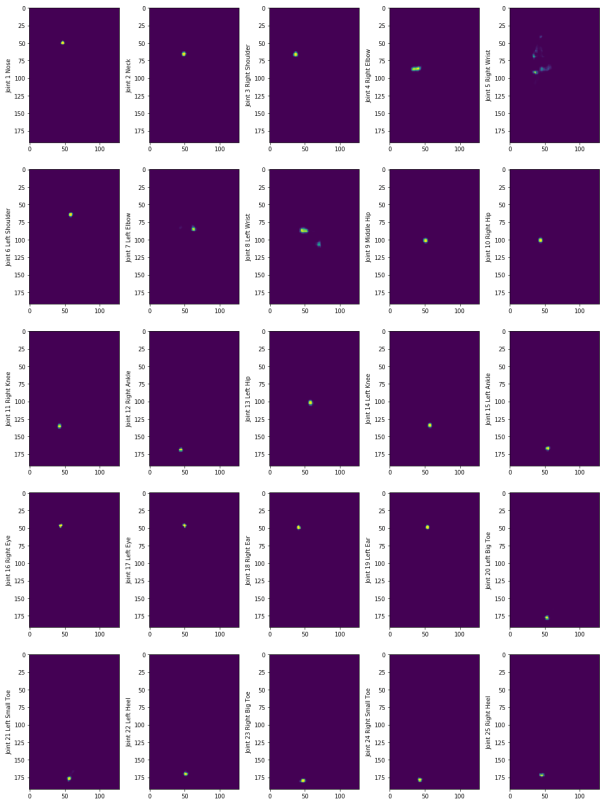

Fig. 19. Softmax outputs in each channel (each channel corresponds to a joint)

The latter has very low cross entropy, since the real values between 0 and 1 instead of just ones such as in the former model are used during loss calculation. We continue with Mask+Softmax model, because it has a higher training and validation dice coefficient. Also the joint prediction performance of these two models are similar.

After choosing the model, we checked the softmax outputs at each channel which corresponds to a certain joint. Outputs for a arbitrary image can be seen in Figure 19. We can infer that the model can indeed learn the specific joint locations by maximizing the softmax output at certain pixels.

After we determined the softmax outputs for each joint, we intended to draw limbs between neighbour joints. We take the pixel that maximizes the softmax output for each channel (joint), and we draw lines between those pixels. Neighbour joints are already specified by OpenPose documentation. An example of drawn skeletons (joints & limbs) can be observed in Figure 20.

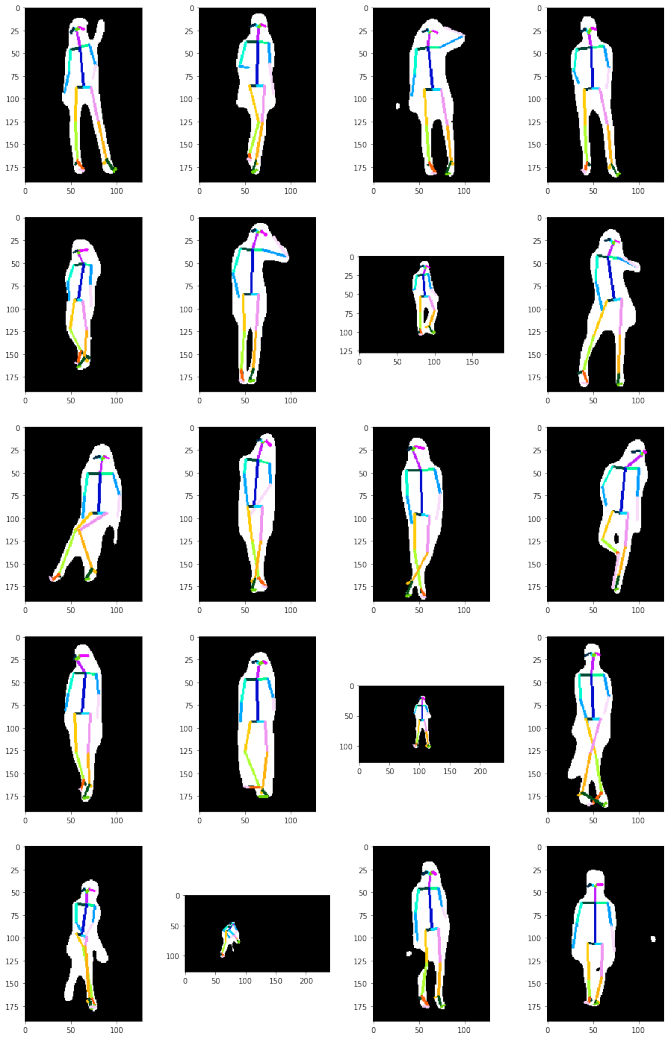

Fig. 20. Predicted skeletons on predicted human pose masks

As a result, we managed to create a model that finds human pose segmentation masks, joints and limbs of a human in a single-person full-body image simultaneously.

IV. Conclusion

We compared the performances of U-Net and U-Net architecture with circular harmonic filters. This project is one of the first projects that use Harmonic Networks in human pose segmentation. We created the target values using OpenPose and Mask RCNN. We found out that U-Net with regular convolutions has a slightly better segmentation loss, and it has a lower training time. Then we continued with U-Net model to create a combined model which can estimate human poses and locate the human joints in a single-person image. We performed four different experiments using U-Net with regular convolutions. Using different target values and loss functions, we managed to build a promising model which can find human poses accurately using a 2-channel (one for background and one for foreground pixels) softmax output and dice loss, and locate the specific joint places using a 26-channel softmax output with categorical cross entropy simultaneously. Also, we could draw the limbs between the predicted joint centers, and we presented the skeletons on estimated human poses. Moreover, we found out that we have sharper and more accurate human pose masks if we predict human pose masks and joint locations together.

Since we did not have enough time to finish the project, we did not change the weights of loss functions and check if we have better results with that. One might have a better combined model by changing the loss weights. Also, we were mostly limited during model creation, because of GPU's memory. With a GPU which has more memory than 12 GB may help building bigger models, and providing more accurate masks and joint locations. Nevertheless, we built a combined model which can estimate human poses, detect joint locations and draw skeletons on mask outputs concurrently. Also our Harmonic network model is one of the first models that was used for a human pose segmentation task. In future, we think that the model we built can be used for a human weight and height prediction task.

Acknowledgment

This semester project is conducted under the supervision of Radhakrishna Achanta and Dorina Thanou at Swiss Data Science Center.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[2] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[4] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5028–5037.

[5] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation." in *BMVC*, vol. 2, no. 4, 2010, p. 5.
[6] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.
[7] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan, "Human parsing with contextualized convolutional neural network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1386–1394.
[8] "Reimplementation of harmonic networks in pytorch," https://github.com/jatentaki/harmonic, accessed: 2019-06-10.
[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[10] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.