



Automated Hand Joint Classification of Psoriatic Arthritis Patients Using Routinely Acquired Near Infrared Fluorescence Optical Imaging

Lukas Zerweck^{1,3}(✉) , Stefan Wesarg^{2,3} , Jörn Kohlhammer^{2,3} ,
and Michaela Köhm^{1,3,4} 

¹ Fraunhofer Institute for Translational Medicine and Pharmacology ITMP,
Frankfurt am Main, Germany

lukas.zerweck@itmp.fraunhofer.de

² Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany

³ Fraunhofer Cluster of Excellence Immune-Mediated Diseases CIMD,
Frankfurt am Main, Germany

⁴ Division of Rheumatology, Goethe-University Frankfurt,
Frankfurt am Main, Germany

Abstract. Near infrared fluorescence optical imaging (NIR-FOI) is a relatively new imaging modality to diagnose arthritis in the hands. The acquired data has two spatial dimensions and one temporal dimension, which visualizes the time dependent distribution of an administered color agent. In accordance with previous work, we hypothesize that the distribution process allows a joint-wise classification into *inflammatory affected* and *unaffected*.

In this work, we present the first approach to objectively classify hand joint NIR-FOI image stacks by designing, training, and testing a neural network. Previously presented model architectures for spatio-temporal classification do not yield satisfying results when trained on NIR-FOI data. A recall value of 0.812 of the over- and a recall value of 0.652 of the underrepresented class is achieved, the model's robustness tested against small variations and its attention visualized in activation maps. Even though these results leave room for further improvement, they also indicate, that the model architecture can capture the latent features of the data. We are confident, that more available data will lead to a robust classification model and can support medical doctors in using NIR-FOI as a diagnostic tool for PsA.

Keywords: Near Infrared Fluorescence Optical Imaging · Spatio-Temporal Data Classification · Neural Network · Psoriatic Arthritis

1 Introduction

Psoriasis arthritis (PsA) affects 0.3-1% of the general population [3] and may lead to permanent structural joint damage [2] if left untreated. An early diagno-

sis and start of treatment can slow down disease progression and can reduce costs [2]. However, clear indicators of early stages for diagnosing PsA, e.g. biomarkers, are yet to be discovered.

In recent years, near infrared fluorescence optical imaging (NIR-FOI) emerged as a potential diagnostic tool in the field of rheumatology to detect the existence and early stages of arthritis in the hand joints and is, especially for PsA, part of ongoing research. However, up until today, the acquired data is evaluated semi-quantitatively by capturing each investigated joint’s inflammatory status in a NIR-FOI specific score, from which FOIAS is most frequently used [10].

In this work, we present the first approach using a neural network for an objective evaluation of NIR-FOI spatio-temporal imaging data by extracting an image stack per joint and classifying the entire three-dimensional spatio-temporal data stack into *inflammatory affected* and *unaffected*.

We use the FOIAS score, which directly evaluates the imaging data, as ground truth label to show, that the suggested model architecture can capture the latent features of the given data. In future work, we will tackle further research questions, of which the early detection of PsA is the main focus.

2 Background

Related Work. In this work, we present our idea for NIR-FOI joint stack classification, which is described in detail in Subsect. 3. There are different approaches for classifying spatio-temporal data in the literature. We mention the approaches we compared to the suggested idea, but do not present comparison results.

Due to the temporal connection between different slices, the approaches are often based on recurrent neural networks (RNN). Two RNN based approaches, which are used for video classification, are gated recurrent units (GRU) [1] and Long Short-Term Memory units (LSTM) [6]. The usage of GRUs is suggested by the official Keras website [9], while in Halder et al. [4] a CNN-BiLSTM is presented to capture the temporal relations. In Mao et al. [8] a graph network for video classification is proposed. However, the data in Mao et al. [8] differ from ours in the sense that an overall video semantic, based on different camera angles and views, needs to be found. We implemented all these approaches and compared them to the presented model in this work. However, none of these yielded satisfying classification results.

Thus, we divided our model into two simpler calculations. Extraction of a temporal embedding, which is then classified by a convolutional neural network. Using a temporal convolutional network [7] for the time embedding extraction, lead to better results than the RNN based approaches, but does not outperform our model.

Since all of the approaches mentioned above showed good results in the past, we currently do not have a satisfactory explanation for the low performance classifying the data used in this work. One difference to previously suggested models is the low resolution of the used data for this work.

Data. NIR-FOI is an imaging modality tailored to the color agent indocyanine green. It acquires 360 images at one image per second, resulting in two spatial dimensions (x, y) and one temporal dimension (t). Even though a general distribution process is described in previous work [10], the color agent distribution varies greatly between data sets. Thus, a neural network is trained to learn the latent features of *affected* and *unaffected* joints.

The data sets used for this work were acquired during a multi-centre study, which fulfilled Good Clinical Practice Guidelines in accordance with the Declaration of Helsinki. It was approved by the ethics committee of the University Hospital Frankfurt am Main and all participants gave signed consent to be included in the study and to the usage of their data for research purposes. In total, 659 data sets from 27 patients, with 104 joints labeled as *affected* and 555 joints labeled as *unaffected*, are included.

3 Method

All described methods are implemented using Python 3.3.8 and Tensorflow 2.3.0. To train the neural network for classifying joints as *affected* and *unaffected*, each data set needs to go through a pre-processing pipeline.

Firstly, 26 joint stacks are extracted from each patient’s image stack, based on a calculated two-dimensional segmentation map. Each stack is adjusted to a size of width = 50, height = 50, and channel = 360 by spatial interpolation. The wrists and interphalangeal joint of both thumbs are not considered, due to their different positioning with regard to the measurement device (CCD chip). Then, each stack goes through a sequence of three steps for each training epoch: Addition of white noise, z-score normalization, and further augmentation (e.g. random flipping, random rotation, and more).

To test the neural network’s robustness, during model testing, described in Subsect. 4, white - and salt-and-pepper noise are added to the data sets.

Model. The entire model structure is visualized in Fig. 1. The model has two main paths. While one side, including the “time blocks” and “space blocks”, is learning the latent features (latent path), the other path is serving as a big skip connection (inspired by a residual block [5]) (skip path).

For the latent path, firstly five “time blocks” with a decreasing number of channels is performed. In each “time block” a one-dimensional convolution is calculated, which serves as a fully connected layer in temporal dimension. The output of each “time block” is concatenated to a $50 \times 50 \times 5$ latent space: The time embedding. Continuously decreasing the number of channels, while keeping each step’s embedding, combines the high- and low-level time features. Then, the time embedding runs through three “space blocks”, to capture the time embedding’s spatial features. The final tensor of shape $12 \times 12 \times 512$ is reduced to a 512-element vector by global average pooling to be concatenated with the result of the skip path.

The idea of the skip path is to capture the input data set in a few values and infuse it into the fully connected layers at the end of the model, to make learning

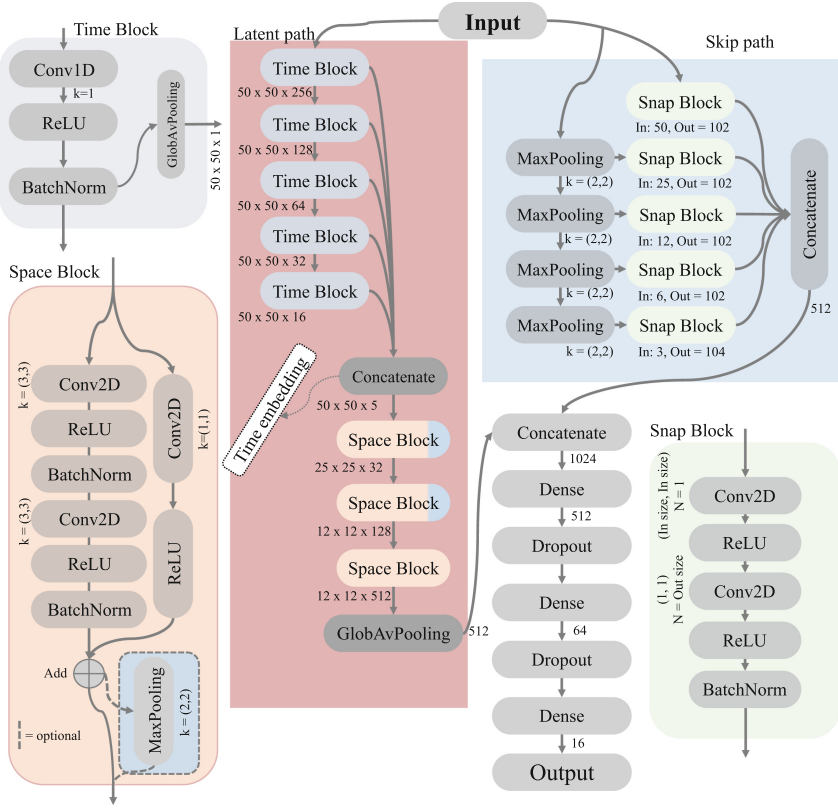


Fig. 1. Model structure. The figure visualizes the entire model architecture with its two main paths: The latent path capturing latent features in temporal and spatial dimension, and the skip path to make learning easier. Additionally, the different block types are shown (Time-, Space- and Snap Block).

easier. It consists of five “snap blocks”, which all have the original input in its initial or spatially pooled dimensions. A “snap block” captures the whole stack in a single value by performing a two-dimensional convolution, with the filter size as big as the spatial dimensions and *valid* padding. This single value is convoluted with a 1×1 filter either 102 or 104 times, to adjust the vector size. The five results of the “snap blocks” are concatenated to a vector of size 512 and then concatenated with the latent path output.

Finally, the concatenated vector of size 1024 runs through four dense layers with decreasing size and two dropout layers.

Model Training. The available data is randomly split with a ratio of 70% (abs.: 416) training -, 20% (abs.:132) validation - and 10% (abs.: 66) test data, while considering the class imbalance. However, the validation data is only used

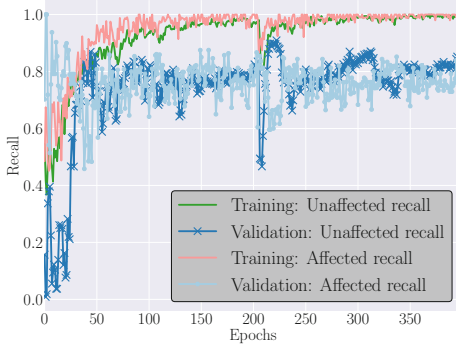


Fig. 2. Recall for trainings- and validation data for both classes and all epochs.

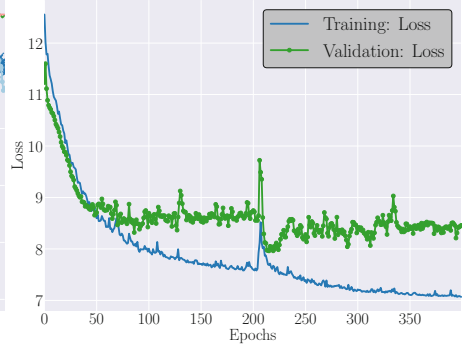


Fig. 3. Trainings- and validation loss for all epochs.

for visualizing the model’s performance on non-training data during training and is not used for fine tuning. Thus, validation and test data can be considered equal, during model evaluation. The model is trained with the following setup: Epochs: 400, batch size: 64, optimizer: Adam, loss: binary cross-entropy. The learning rate is not stable and decreases over time starting at 10^{-4} and ends at $2 \cdot 10^{-5}$ with two intermediate increases.

Model Robustness Testing. As mentioned before the amount of available data sets for the class *affected* is relatively small. Thus, different combinations of training- and test data sets can lead to very different results. To get a reliable result, the neural network is trained multiple times from scratch with the same training conditions but with a random assignment of samples to training - and test data set.

4 Results

The results of all training runs are summarized in Table 1. The model with the gray background is used for further investigation. Due to the high class imbalance, the accuracy, as well as precision value, have limited explanatory power. Instead, the recall value for both classes over all 400 epochs is visualized in Fig. 2. Additionally, the loss is visualized in Fig. 3. In both figures, the increase in learning rate causes a fluctuation in recall and loss value. However, the recovery of both values after a small amount of epochs indicates, that the model is optimized towards the global and is not stuck in a local minimum. The recall and loss values are relatively stable for the validation data, which indicates a maximal performance with the given data. Additionally, the validation loss does not increase after many epochs of training. Thus, no numerical signs of overfitting can be observed.

Table 1. Performance of all trained models on test and validation data.

	True prediction		False prediction		Recall		Precision	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
Model 1	124	20	43	11	0.743	0.645	0.919	0.317
Model 2	132	24	35	7	0.790	0.774	0.950	0.407
Model 3	142	18	25	13	0.850	0.581	0.916	0.419
Model 4	138	17	29	14	0.826	0.548	0.908	0.370
Model 5	144	23	23	8	0.862	0.742	0.947	0.500
Model 6	143	21	24	10	0.856	0.677	0.935	0.467
Model 7	130	22	37	9	0.778	0.710	0.935	0.373
Model 8	128	19	39	12	0.766	0.613	0.914	0.328
Model 9	140	18	27	13	0.838	0.581	0.915	0.400
	Σ	Σ	Σ	Σ				
	1221	182	282	97	0.812	0.652	0.926	0.392

Investigating Model Results. To get a deeper understanding of the trained neural network, two evaluation steps are performed. Firstly, the model’s focus on the input joint stacks is visualized. Secondly, its robustness against small data changes is tested.

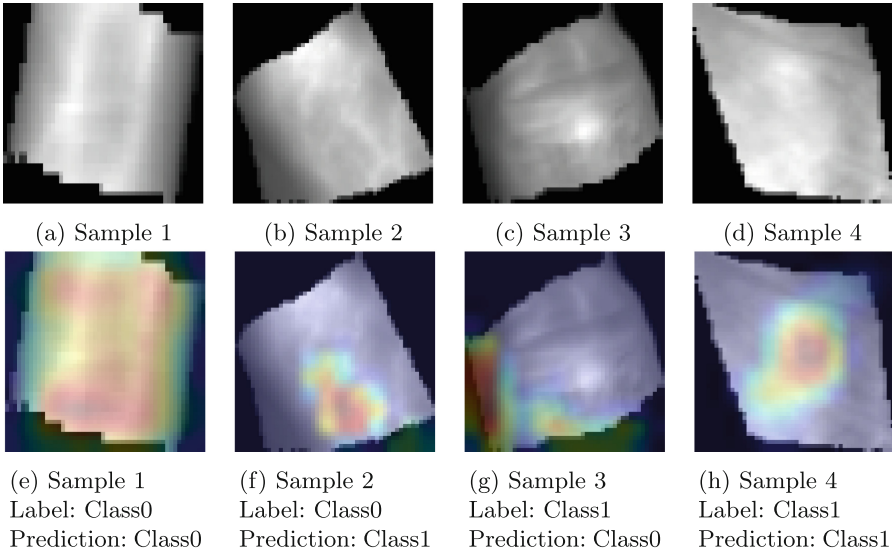


Fig. 4. For the images in (a) - (d) the standard deviation along the temporal dimension is calculated. (e) - (h) show the overlay of these stand deviation images with the corresponding activation map. A sample for each possible classification result is visualized.

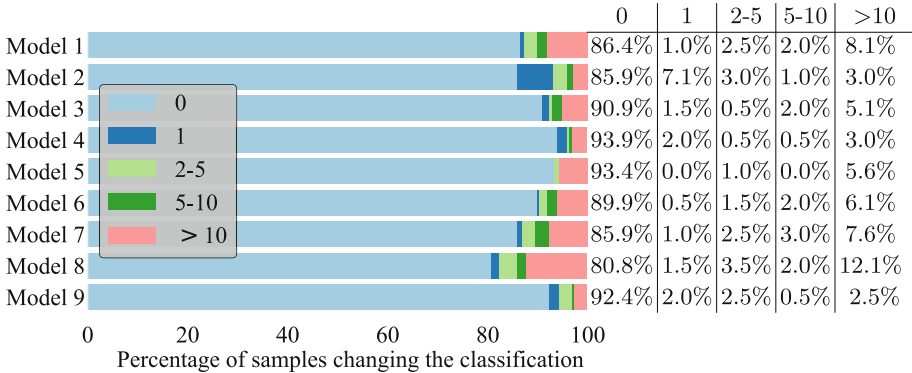


Fig. 5. Summary of the robustness test. Each column contains the percentage of data sets, for which the classification is **switched** as many times as the bar color or column title states (e.g. the label/column 0 displace the percentage of data sets, which switch 0 times the classification during all 100 classifications).

For four samples activation maps are shown in Fig. 4, to visualize the influence of different parts of the initial joint stack on the classification. To capture the dynamic distribution progress in one image and thus, enable an evaluation of the activation maps, the standard deviation along the temporal dimension is calculated, and visualized in Fig. 4 (a)-(d). While the neural network focuses on meaningful parts for correctly classified samples (Fig. 4 (e) and (h)), it seems to focus on less meaningful areas for wrongly classified samples (Fig. 4 (f) and (g)). Especially, in Fig. 4 (g) the model seems to miss the bright spot and rather focuses on the joint’s edge.

To test the validity of the final result, all test and validation data sets are predicted 100 times, with randomly added white - and salt-and-pepper noise individually for each sample and test. A sample switching between different predictions due to added noise represents an uncertain neural network and can indicate overfitting (the approach is inspired by adversarial attacks). The smallest number of 0 switches can be observed for model 8 with a switch rate of 80.8%. Model 5, the one highlighted in Table 1, has a switch rate of 93.4% and thus, appears robust against smaller changes in the data (the results are summarized in Fig. 5). Both evaluation steps support the conclusion of a not overfitted model.

5 Discussion and Future Work

In this work, we present a neuronal network architecture, to classify joints, based on NIR-FOI spatio-temporal imaging data. Approaches from the literature to classify the data used in this work do not yield satisfying results, for which we currently do not have a substantiated explanation.

With the presented idea, an average recall of 0.812 of the over- and an average recall of 0.652 of the underrepresented class is achieved. These values clearly

state, that the trained model performs better than random guessing, especially considering the class imbalance, but still has a lot of potential for improvement. The development over all epochs of both recall values, as well as the loss, indicate, that a learning is achieved, without overfitting the model. This conclusion is also supported by the repetitive testing, for which the vast majority of samples do not change their classification when applying minor changes to the input image (compare Fig. 5).

The visualized activation maps, shown in Fig. 4, allow the conclusion that the model can capture the latent features of the data. While the activation maps of correct classified samples support this conclusion (Fig. 4 (e) and (h)), the wrongly classified samples also indicate that more data is necessary to eliminate wrong attention (Fig. 4 (f) and (g)).

In summary, the presented pipeline shows the potential to be a robust classifier, for the extracted NIR-FOI joint stacks. Thus, the presented neural network architecture will be trained on more data to develop an automated evaluation system to support clinicians in diagnosing PsA with the support of NIR-FOI.

Acknowledgment. This project has received funding from the Innovative Medicines Initiative 2 Joint Undertaking (JU) under grant agreement No 101 007 757 (Hippocrates). The JU receives support from the European Union’s Horizon 2020 research and innovation program and EFPIA.

References

1. Ballas, N., Yao, L., Pal, C., Courville, A.: Delving deeper into convolutional networks for learning video representations. arxiv.org/abs/1511.06432 (2015)
2. D’Angiolella, L.S., et al.: Cost and cost effectiveness of treatments for psoriatic arthritis: a systematic literature review. *PharmacoEconomics* **36**(5), 567–589 (2018). <https://doi.org/10.1007/s40273-018-0618-5>
3. Gladman, D.D.: Psoriatic arthritis: epidemiology, clinical features, course, and outcome. *Ann. Rheum. Dis.* **64**(suppl 2), ii14–ii17 (2005). <https://doi.org/10.1136/ard.2004.032482>
4. Halder, R., Chatterjee, R.: CNN-BiLSTM model for violence detection in smart surveillance. *SN Comput. Sci.* **1**(4) (2020). <https://doi.org/10.1007/s42979-020-00207-x>
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
7. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. arxiv.org/abs/1611.05267 (2016)
8. Mao, F., Wu, X., Xue, H., Zhang, R.: Hierarchical video frame sequence representation with deep convolutional graph network. In: *ECCV 2018 Workshops*, pp. 262–270 (2019). https://doi.org/10.1007/978-3-030-11018-5_24
9. Paul, S.: Video classification with a CNN-RNN architecture (2021). https://keras.io/examples/vision/video_classification/. Accessed 25th Apr 2023

10. Werner, S.G., et al.: Indocyanine green-enhanced fluorescence optical imaging in patients with early and very early arthritis: a comparative study with magnetic resonance imaging. *Arthritis Rheum.* **65**(12), 3036–3044 (2013). <https://doi.org/10.1002/art.38175>