



# Automated bone erosion scoring for rheumatoid arthritis with deep convolutional neural networks

Zurich University of Applied Sciences

**Author:** Janick Rohrbach

**Supervisor:** Prof. Dr. Oliver Dürr  
Prof. Dr. Beate Sick

**Industrial Partner:** Seantis GmbH

**External Supervisor:** Fabian Reinhard  
Dr. Tobias Reinhard

**Date:** December 22, 2017

## **Declaration of originality**

### **Project Thesis at the School of Engineering**

By submitting this project thesis, the undersigned student confirms that this thesis is his own work and was written without the help of a third party.

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the project thesis have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

Zurich, December 22, 2017

Janick Rohrbach

# Abstract

Rheumatoid arthritis can cause irreversible damage to the joints. The severity of these bone erosions is scored by using x-ray images. This is usually done by a trained rheumatologist or radiologist and takes several minutes per patient.

This thesis shows a method to automatically score the joints in x-ray images with deep convolutional neural networks. We take a classification and a regression approach on x-ray images of joints from the left hand. In the classification task we predict the Ratingen-score on a discrete integer scale from 0 to 5. The model achieves class normalized validation and test accuracies of 42 % and 43 % respectively. The class normalized accuracies of predictions that are off by no more than 1 are 82 % for the validation set and 83 % for the test set. The regression model predicts the continuous percentage of bone erosion between 0 % and 100 % with a validation and test mean squared error of 72.8 and 97.6 respectively. The mean absolute error is 3.1 for the validation set and 3.5 for the test set.

An automated scoring of bone erosion could help rheumatologists to spend less time with the scoring and have more time with the patient.

## Acknowledgements

I would like to express my sincere thanks to my supervisors Oliver Dürr and Beate Sick who provided me with guidance and support during the writing of this thesis.

I would also like to thank Fabian Reinhard and Tobias Reinhard (Seantis GmbH) for their valuable inputs.

Further, I want to acknowledge the SCQM foundation which made this thesis possible by providing the comprehensive dataset. A list of rheumatology offices and hospitals that are contributing to the SCQM registries can be found on [www.scqm.ch/institutions](http://www.scqm.ch/institutions). The SCQM is financially supported by pharmaceutical industries and donors. A list of financial supporters can be found on [www.scqm.ch/sponsors](http://www.scqm.ch/sponsors).

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Background . . . . .	6
1.2	Related literature . . . . .	6
1.3	Aim and scope of this thesis . . . . .	7
<b>2</b>	<b>Theory</b>	<b>8</b>
2.1	Finger joints . . . . .	8
2.2	Ratingen-score . . . . .	8
2.3	Rau-score . . . . .	8
2.4	Disease activity score (DAS28) . . . . .	9
2.5	Artificial neural networks . . . . .	10
2.5.1	Fully connected neural networks . . . . .	11
2.5.2	Convolutional neural networks . . . . .	11
<b>3</b>	<b>Data</b>	<b>13</b>
3.1	Data preparation . . . . .	14
<b>4</b>	<b>Methods and results</b>	<b>16</b>
4.1	Software and infrastructure . . . . .	16
4.2	Base models . . . . .	16
4.2.1	Classification model . . . . .	16
4.2.2	Regression model . . . . .	19
4.3	Transfer learning . . . . .	23
4.4	Model evaluation . . . . .	25
4.5	Comparing the classification and regression model . . . . .	26
4.6	Attention maps . . . . .	28
4.7	Analysis of the embeddings . . . . .	28
4.7.1	K-nearest-neighbor of embeddings . . . . .	28
4.7.2	Analysis of outliers in the embeddings . . . . .	29
4.8	Analysis of correlations between bone erosion and disease activity . . . . .	30
<b>5</b>	<b>Discussion</b>	<b>32</b>
<b>6</b>	<b>Conclusion and outlook</b>	<b>34</b>

# 1 Introduction

This thesis shows a method for the automated scoring of x-ray images of patients with rheumatoid arthritis.

## 1.1 Background

Rheumatoid arthritis is an autoimmune disease. Which means that the disease is caused by a malfunctioning immune system. The immune system attacks healthy tissue instead of bacteria and viruses. This causes inflammation in the joints. Irreversible damage to the bone in the joint can occur, if the inflammation lasts for a long time. [1] Rheumatoid arthritis is incurable, merely the symptoms can be treated.

Today, the severity of the bone erosion is assessed by a trained rheumatologist by using x-ray images of hand and feet. This process takes several minutes per patient. This thesis shows, how recent advances in computer vision make it possible to automate this task. This leads to time savings which in return help the rheumatologist to spend more time with the patient.

The Swiss Clinical Quality Management in Rheumatic Diseases (SCQM) Foundation runs a national registry of inflammatory rheumatic diseases. [2] They have collected anonymized patient data for over 10 years and provide the x-ray images used for this analysis.

Seantis GmbH, the industrial partner for this thesis, is a Swiss company that develops data driven web applications for medical research, public administration and aviation. [3] For their customer SCQM they want to automate the bone erosion assessment. They already have a working algorithm, which detects the body part shown in the x-ray image. A second algorithm detects the joints in the image and extracts them as single images. These images are then used together with the bone erosion scores to train our model.

## 1.2 Related literature

There are several applications where convolutional neural networks are used in medical research.

A recent paper from Tajbakhsh et al. [4] investigated whether fine-tuning a pre-trained CNN is better than training a CNN from scratch when applied to medical images. They find that pre-trained networks with fine-tuning always outperformed or at least performed as well as CNNs trained from

scratch. They further recommend a layer-wise fine tuning which seems to outperform shallow and deep tuning.

A study by Paul et al. [5] tried to classify osteoporosis by considering x-ray images of the bone. This task proved to be very difficult as the x-ray images from healthy patients look very similar to the ones of patients with the disease. By using a transfer learning approach they achieved a validation accuracy of 44.82 %.

Zhou et al. [6] used a two-level ensemble of neural networks to identify lung cancer cells on x-ray images of the chest. The first-level ensemble classifies whether a cell is a cancer cell or not by using full voting. The second-level ensemble is used only on cells classified by the first-level as cancer cells. It differentiates between different cancer classes as well as a non-cancer class. This ensemble works with plurality voting. The authors state that this method achieves a high accuracy and a low rate of false negatives.

A report from Chen [7] showed the application of convolutional neural networks on x-ray images of hands to predict the developmental bone age. He achieves a top one and two accuracy of 46 % and 70 % respectively. This result is close to previously used methods which use manual segmentation and handcrafted features.

In a degree project Hensman and Masko [8] looked at the impact of imbalanced training data for CNNs. They find, that heavy imbalances have a strong impact on the performance and suggest oversampling of minority classes to improve the performance of the network.

### 1.3 Aim and scope of this thesis

The aim of this thesis is to predict bone erosion scores from x-ray images. We further examine how the bone erosion and the disease activity are correlated.

The work is based on images of the left hand only. There also exist images of right hands as well as images of left and right feet. But at this point in time, only the joints of left hands have been extracted from the images. It is assumed that the model will perform similar on the joints of the right hand. By fine-tuning the model on the images of joints from feet it should also perform well for those images.

## 2 Theory

### 2.1 Finger joints

Figure 1 shows an x-ray image of a left hand similar to the images received from the SCQM foundation. The five proximal interphalangeal (PIP) joints and the five carpometacarpal (MCP) joints are shown with blue bounding boxes. These are the joints, that are most affected by rheumatoid arthritis. The wrist joint is also affected, but we limited our analysis to the five PIP joints and the five MCP joints. Each of these joints is assessed with a score by a trained rheumatologist or radiologist. The scoring method is described in the next section.

### 2.2 Ratingen-score

The most important criteria for the effectiveness of a treatment is the influence on the radiological progression. To quantify the irreversible bone erosion in the joint, several scoring methods were developed. The score used in this thesis is called Ratingen-score, it estimates the percentage of eroded joint surface. [9]. The labels of our data lie within 0 and 100 and correspond to the percentage of joint surface erosion. These values can easily be converted to Ratingen-Scores according to Table 1.

Stage	Description
0	Normal joint
1	One or more erosions, less than 20 % of the joint surface is eroded
2	21 % - 40 % of the joint surface is eroded
3	41 % - 60 % of the joint surface is eroded
4	61 % - 80 % of the joint surface is eroded
5	More than 80 % of the joint surface is eroded

Table 1: Disease stages of the Ratingen-score [9]

### 2.3 Rau-score

This score is an overall score, which is calculated from the individual Ratingen-scores. The sum of the Ratingen-scores for all 32 joints (5 PIP, 5 MCP and

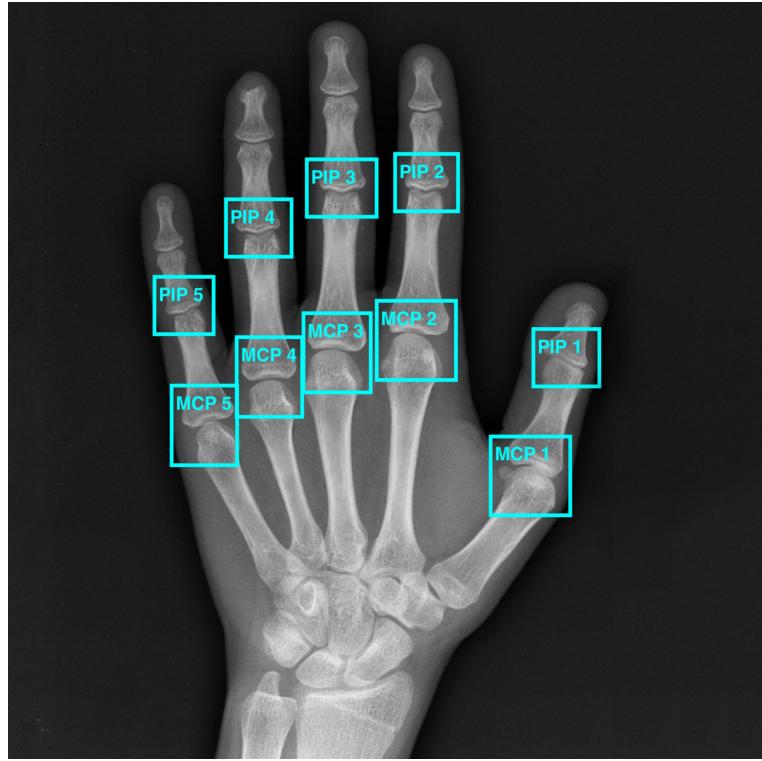


Figure 1: Proximal interphalangeal (PIP) joints and carpometacarpal (MCP) joints of the left hand. Cropped images of these ten joints were used to train the neural networks.

Original image by Nevit Dilmen (CC BY-SA) [https://commons.wikimedia.org/wiki/File:Medical\\_X\discretionary{-}{}{}Ray\\_imaging\\_OP06\\_nevit.jpg](https://commons.wikimedia.org/wiki/File:Medical_X\discretionary{-}{}{}Ray_imaging_OP06_nevit.jpg)

1 wrist joint per hand and 5 joints per foot) is multiplied by 38 and divided by the number of scored joints.

## 2.4 Disease activity score (DAS28)

The disease activity score (DAS28) measures the disease activity for the following 28 joints. 5 PIP, 5 MCP and 1 wrist joint per hand plus elbow, shoulder and knee joints [10]. The score is derived from the following four measurements.

- (a)  $n_s$  = Number of swollen joints

- (b)  $n_t$  = Number of tender joints
- (c)  $ESR$  in mm/hr or  $CRP$  in mg/L = Blood markers of inflammation (either the erythrocyte sedimentation rate (ESR) or C reactive protein (CRP))
- (d)  $g_h$  in mm = Patients global assessment of disease activity on a 100 mm long scale from very good (0 mm) to very bad (100 mm).

The DAS28 score is then calculated as follows. [11]

$$DAS28_{ESR} = 0.56 * \sqrt{n_t} + 0.28 * \sqrt{n_s} + 0.7 * \ln(ESR) + 0.014 * g_h$$

$$DAS28_{CRP} = 0.56 * \sqrt{n_t} + 0.28 * \sqrt{n_s} + 0.36 * \ln(CRP + 1) + 0.014 * g_h + 0.96$$

$DAS28 > 5.2$  = high disease activity

$DAS28 < 3.2$  = low disease activity

$DAS28 < 2.6$  = remission

## 2.5 Artificial neural networks

This section offers a very brief introduction to artificial neural networks. A more in-depth explanation can be found in Andrey Karpathy's course notes for the Stanford class CS231n. [12]

The structure of an Artificial neural network (ANN) is inspired by the human brain. A brain consists of approximately 100 billion neurons which form an interconnected network. The neurons can communicate with each other by transmitting electrical potential. If the potential in a neuron reaches a certain threshold, it fires and transmits the potential to connected neurons. [13]

An ANN is a very simplified model of this biological process. A single neuron can be described by the following equation.

$$f \left( \sum_i w_i * x_i + b \right)$$

Where  $x_i$  are the inputs,  $w_i$  are the weights and  $b$  is a bias term. The activation function  $f$  models the firing of the neuron.

These single neurons can then be combined to networks. The most simple network is the fully connected neural network described in the following section.

### 2.5.1 Fully connected neural networks

A fully connected neural network (FCNN) has an input layer, arbitrarily many hidden layers and one output layer. The neurons of each layer are connected to every neuron of the next layer. The data can only flow in one direction, from the input layer towards the output layer. Figure 2 shows a possible structure for a FCNN.

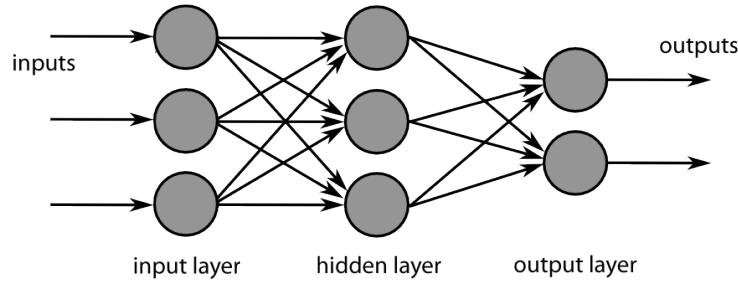


Figure 2: Feed forward neural network with one hidden layer.

Image by Chrislb (CC BY-SA) [https://commons.wikimedia.org/wiki/File:MultiLayerNeuralNetworkBigger\\_english.png](https://commons.wikimedia.org/wiki/File:MultiLayerNeuralNetworkBigger_english.png)

The number of neurons per layer specifies the width of the neural network, whereas the number of hidden layers specifies the depth of a neural network. A neural network with many hidden layers is called a deep neural network.

For supervised learning the weights and biases of this network can be trained by using back-propagation. The input is fed through the network with randomly initialized parameters. The output is then compared to the true values by using a loss-function. The loss is then back-propagated through the network to adjust the parameters. With every training step, this process is repeated and the loss decreases. This process is also called learning. And for deep neural networks we speak of deep learning.

A special type of the FFNN, used for image recognition, is the convolutional neural network, which is described in the next section.

### 2.5.2 Convolutional neural networks

Convolutional neural networks (CNNs) take an image as an input. The image can be seen as a 3-dimensional matrix, where the third dimension includes the different color channels. Instead of fully connected layers, convolutional

layers are used. Convolutions work as filters that detect different features in the image. These filters usually have a small size (e.g. 3x3) and are moved over the image. Figure 3 shows a possible architecture of a CNN with multiple convolutional layers and a fully connected layer at the end.

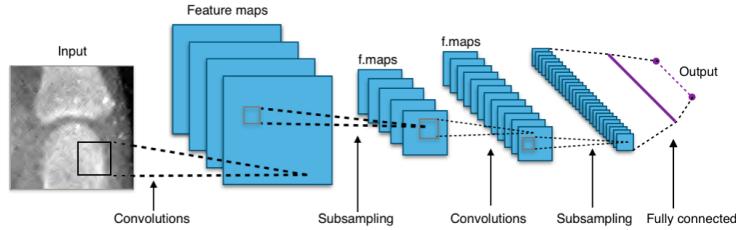


Figure 3: Structure of a convolutional neural network.

Original image by Aphex34 (CC BY-SA) [https://commons.wikimedia.org/wiki/File:Typical\\_cnn.png](https://commons.wikimedia.org/wiki/File:Typical_cnn.png)

In contrast to classical machine learning we don't need to extract any features beforehand. The hidden layers of the deep convolutional neural network will do the feature extraction automatically.

### 3 Data

The received data consists of jpg images of the joints and two csv datasets. The main dataset includes the following relevant columns shown in Table 2.

Column name	Description
id_x	Unique observation id
patient_id	Unique patient id
date_x	Date of the consultation
date_y	Date on which the joints were scored
sop_iuid	Unique x-ray image id
body_part	Left/right hand/foot or both hands/feet
hand_left_x	Percentage of bone erosion for joint x
rau_score	Overall Rau-score described in subsection 2.3

Table 2: Columns of main dataset

This is the dataset where the percentages of bone erosion for the joints of the left hand were extracted. The corresponding x-ray image can be found by using the sop\_iuid.

The secondary dataset contains additional scores which only exist for some of the patients and some consultations. A description of the relevant columns is shown in Table 3. The two datasets can be merged on patient\_id and date/date\_x.

Column name	Description
patient_id	Unique patient id
date	Date of the consultation
physician_global_disease_activity	Medical global assessment of disease activity
global_patient_estimate_disease_activity	Patient estimate of disease activity
das28bsr_score	DAS28BSR as described in subsection 2.4
das28crp_score	DAS28ERP as described in subsection 2.4

Table 3: Columns of secondary dataset

The images of the joints were already extracted from all the x-ray images of left hands. In total there were 102'265 images of single joins available.

### 3.1 Data preparation

The data preparation step brings the jpg images into a suitable format that can be used as an input for the CNN.

The original images have values between 0 and 255. We divided the data by 255 in order to have values in the range of [0,1].

The data was randomly split into a training set (70 % of the data), a validation set (20 % of the data) and a test set (10 % of the data). It was split such that all images of the same patient are in the same set.

The images of the joints have varying exposure. Some images are very dark while others are very bright. It was therefore considered to apply a histogram equalization, which is a linear transformation that maps the lightest pixel to 255 and the darkest pixel to 1. However, this transformation did not improve the accuracy of the model and was not used for the final model.

The bone erosion scores of the labeled joints are highly imbalanced. As seen in Figure 4 most of the joints are healthy and received a score of 0. There are quite a few observations with little bone erosion with scores between 0 and 25. There are very little observations with scores higher than 25. Only the fully eroded joints (score = 100) seem to be a bit more frequent.

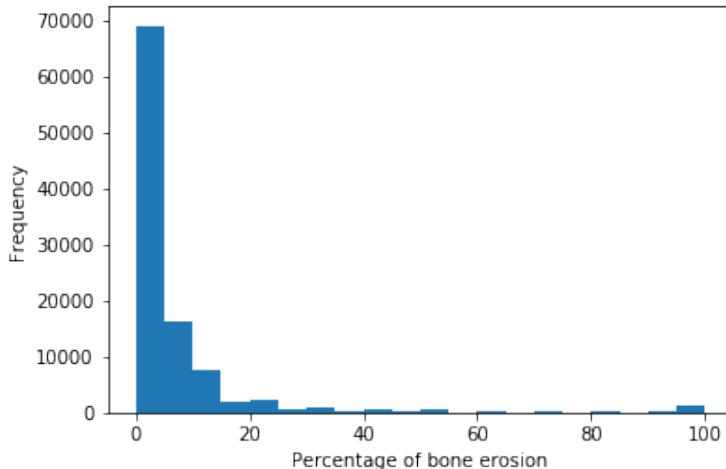


Figure 4: Histogram of the bone erosion scores

When training the CNN, it minimizes the overall loss-function. For imbalanced data the CNN performs bad for the underrepresented part of the data. In this case, the model would be bad predictor for high scores. In order

to make the model a good predictor for all cases, we tried weighting the loss function as well as oversampling the underrepresented classes.

## 4 Methods and results

In order to predict the bone erosion scores, different models were evaluated. This section describes the different steps in the evaluation process. It shows a classification model which predicts the Ratingen-score and a regression model which predicts the percentage of bone erosion.

### 4.1 Software and infrastructure

The models were built in Python 3.5 [14] with the package Tensorflow 1.4 [15] using the high level API Keras [16]. For general machine learning the package Scikit-learn [17] was used. The package Matplotlib [18] was used to produce figures. In addition, the programming language R [19] was used for further analysis of the results and for producing figures.

For the training of the models a Nvidia Titan X (Pascal) graphics card was used.

### 4.2 Base models

We decided to create a classification as well as a regression model. The architecture of both models is similar, each model has 6 blocks of two convolutional layers of the size 3x3 followed by a max pooling layer. The number of filters per convolutional layer is increasing with every second block, whereas the size of the layers is decreasing due to the max pooling layers. Every convolutional layer uses batch normalization before the ReLu activation function.

This first part is identical for both models. We then flatten the output of the last convolution block and use two dense layers with batch normalization, ReLu activation and dropout. Only the number of neurons in the dense layers is different between the two models. The output layer however is different between the two models and described in the next two sections.

#### 4.2.1 Classification model

The classification model directly predicts the Ratingen-score. The output layer has 6 neurons according to the 6 Ratingen classes (0-5). The Softmax activation function is then used to predict the probabilities for each class. The architecture of this CNN is shown in Figure 5.

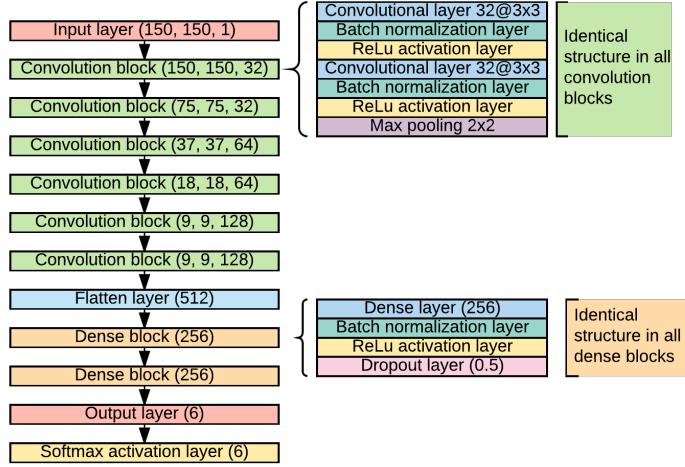


Figure 5: Architecture of the CNN for classification

The model is trained for 25 epochs with the ADAM optimizer and a learning rate of 0.001. Categorical loss entropy is used as the loss function. The images are fed through the model in batches of 100 images at a time.

As described in subsection 3.1 we tried different approaches to handle the imbalanced data.

- (a) The first model is trained on the original data and is used as a basis to compare the results of the other models.
- (b) For the second model we oversampled the underrepresented classes in order to have roughly the same number of images per class. The oversampled images are augmentations of the original data. Random rotations of up to 25 degrees, shearings of up to 0.1 radians and and zooms of up to 10 % were used.
- (c) The third model used the same data as the first model, but the loss function was weighted with class weights. The class weights where chosen such that the class weight multiplied by the number of observation in that class equals to the same value for all classes.

Model b) was trained for the full 25 epochs, whereas the other two models started overfitting and had to be stopped early after 9 epochs. The training

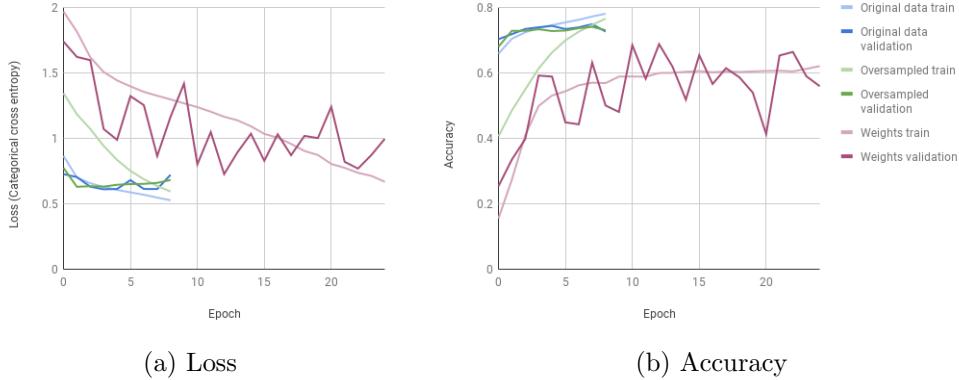


Figure 6: Training and validation loss and accuracy for the three classification models.

and validation loss and accuracy for the three models are shown in Figure 6. Figure 7 shows the normalized confusion matrices for the validation dataset for the three models.

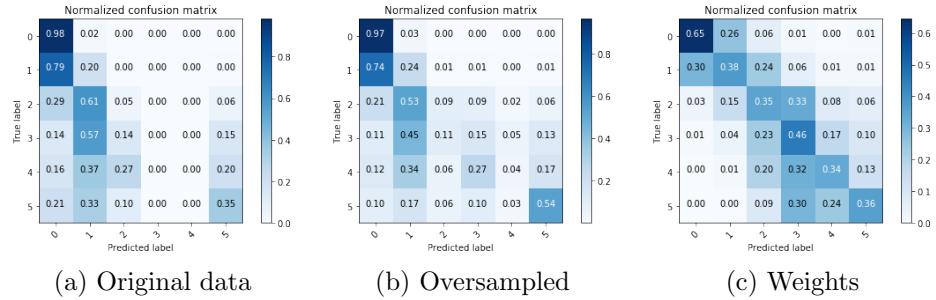


Figure 7: Normalized confusion matrices for the predictions of the classification models on the validation set.

The evaluation metrics for the three models can be seen in Table 4. It turned out that the accuracy is not a suitable metric. Model a) has the highest accuracy but makes a lot of misclassifications in the under-represented classes. Whereas the accuracy of model c) is the lowest, but the accuracy for the underrepresented classes is much better.

Therefore, we decided to use the normalized accuracy as our evaluation metric. It is the mean of the six class accuracies. It describes much better,

whether our model is a good predictor for all classes.

As a second metric, we calculated the normalized accuracy for predictions that are in the right class or no more than one class above or below the correct class. This metric takes into account, that a misclassification by 1 is far less severe than a misclassification by 5.

	a)	b)	c)
Accuracy	0.726	0.73	0.569
Normalized accuracy	0.264	0.337	0.422
Normalized $\pm 1$ accuracy	0.558	0.675	0.817

Table 4: Evaluation metrics for classification.

Considering the class normalized accuracies, model c) is clearly superior to the other two models. Therefore, this model was selected as the base model for the classification task.

#### 4.2.2 Regression model

For the regression model, we decided to predict the cumulative density function (cdf) instead of predicting directly the bone erosion score. The model is only slightly different from the classification model. The output layer has 101 neurons for the discrete integer bone erosion scores from 0 to 100. Since the output layer has much more neurons compared to the classification model, the number of neurons in the dense layers were also increased. The sigmoid activation function is used instead of the softmax activation function. This architecture can be seen in Figure 8.

To train the model, the Continuous Ranked Probability Score (CRPS) was used as the loss function. For the discrete case it is calculated as follows. [20]

$$CRPS = \frac{1}{101 * N} \sum_{n=1}^N \sum_{k=0}^{101} (P(y \geq k) - H(k - R_n))^2$$

Where  $P$  is the predicted distribution,  $N$  is the number of observations,  $R$  is the actual percentage of bone erosion and  $H(x) = \begin{cases} 1 & \text{for } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$  [20]

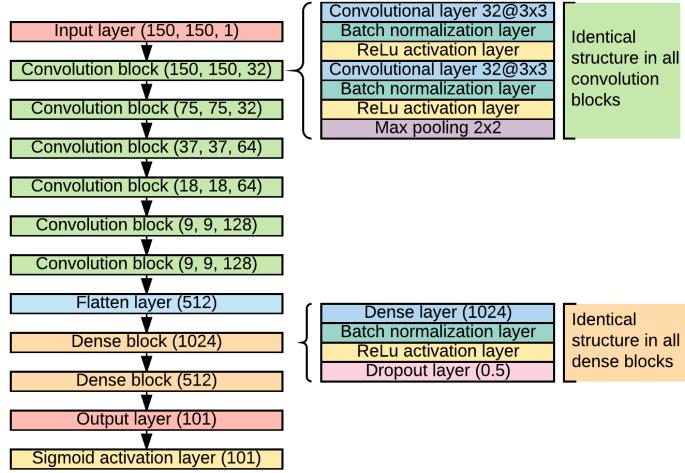


Figure 8: Architecture of the CNN for regression

The blue area visualized in Figure 9 shows approximately what the CRPS is. [20]

The model predicts the discrete cdf for the percentage of bone erosion. An example of a prediction is shown by the blue dots in Figure 10. The green line shows the true label whereas the red line shows the expected value of the predicted distribution. The expected value of our discrete cdf is calculated as follows.

$$E(x) = \sum_{i=1}^{101} (1 - p_i)$$

Where  $p_i$  are the 101 predictions.

This model was trained in the same ways as described in subsubsection 4.2.1 on the original data, oversampled data and with a weighted loss function. This time all three models were trained for 25 epochs, as they did not start overfitting. The train and validation loss and mean absolute error (mae) can be seen in Figure 11. The mae is calculated for the predictions of the cdf and not for the final labels. Figure 12 shows the predictions for the validation set in all three cases.

There is no big visual difference between the three plots. We noticed that the model is not predicting very well for the badly damaged joints with true

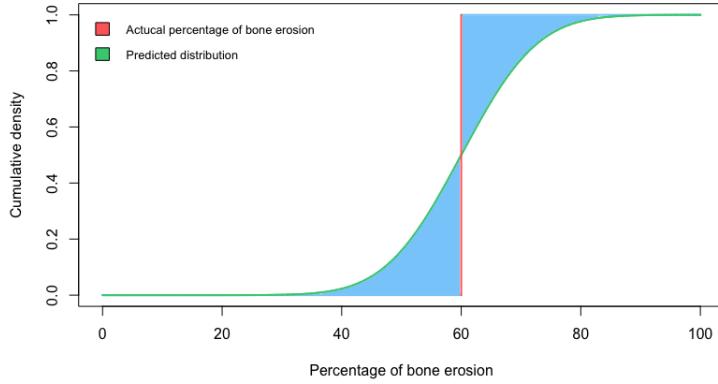


Figure 9: CRPS probability score

labels of 100.

???????????????

In contrast to Figure 11, where mae is calculated for the predictions of the cdf, the mse and mae in the evaluation metrics in Table 5 are calculated for the final predictions. We can see, that model a) has a smallest mean squared error (mse) as well as the smallest mean absolute error (mae) of the three models. Therefore we decided to use model a) as the base model for regression.

	a)	b)	c)
CRPS	0.0291	0.0304	0.0426
Mean squared error (mse)	80.884	83.017	94.488
Mean absolute error (mae)	4.046	4.072	4.057

Table 5: Evaluation metrics for regression.

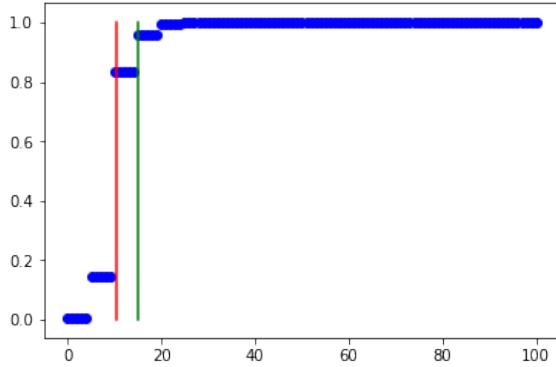


Figure 10: Example of a prediction

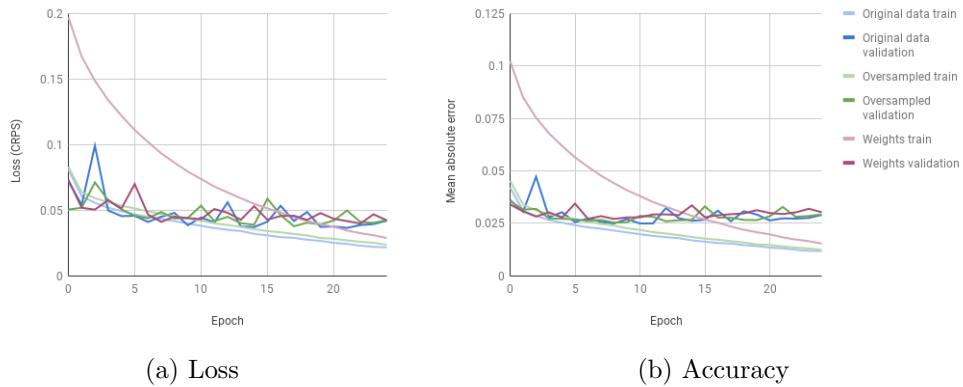


Figure 11: Training and validation loss and mean absolute error for the three regression models.

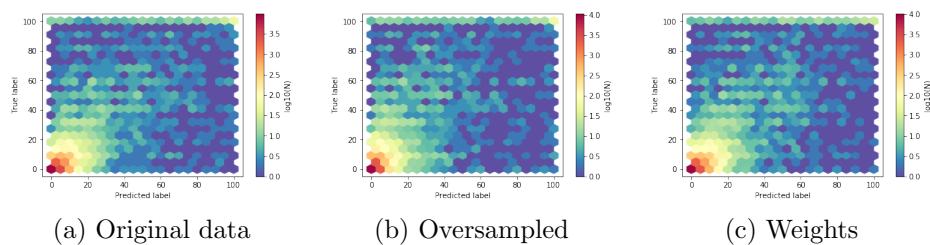


Figure 12: Predictions of the regression models on the validation set.

### 4.3 Transfer learning

As Tajbakhsh et al. [4] suggested in their paper, we wanted to try, whether transfer learning could improve our predictions. We decided to use the Inception V3 model [21] which was pre-trained on the Imagenet dataset. We used the pre-trained weights and cut off the output layer. Instead we added two dense layers and an output layer identical to the output layers described in subsubsection 4.2.1 and subsubsection 4.2.2 for classification and regression respectively. Figure 13 shows the architecture of the two models. Since the model was trained on color images, the greyscale images were converted to RGB. The Inception V3 model further requires the input data to be transformed to [-1,1] which was done in an additional pre-processing step.

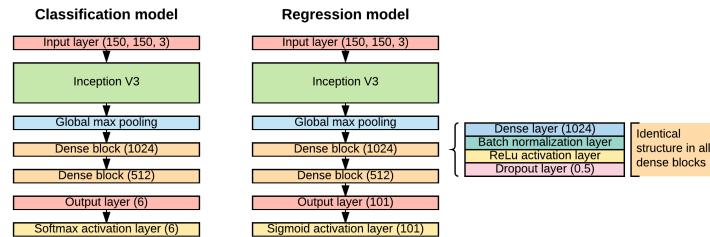


Figure 13: Model architecture for transfer learning

We then froze all existing weights and only trained the dense layers and the output layer for 25 epochs. Afterwards we repeated the training for all layers for an additional 25 epochs. The train and validation loss and mean absolute error are shown in Figures 14 and 15. The predictions for the validation set are shown in Figure 16. When comparing the confusion matrix for the classification model with Model c) in Figure 7 we can see that the transfer learning model has higher accuracy for the extreme cases 0, 1 and 5. However, the model seems to perform worse for the Ratingen-scores 2, 3 and 4. The regression model seems to better predict very bad cases with scores of 100. There are still a few outliers, but far fewer than in the base model.

Table 6 shows the evaluation metrics for the two models. To compare the metrics of the new classification model with the base model, Wilson score intervals on the 5 % significance level were computed. The Wilson score intervals for the normalized accuracy of the base model and the transfer learning model are (0.415, 0.429) and (0.43, 0.443) respectively. The transfer learning model is slightly better. The Wilson score intervals for the second

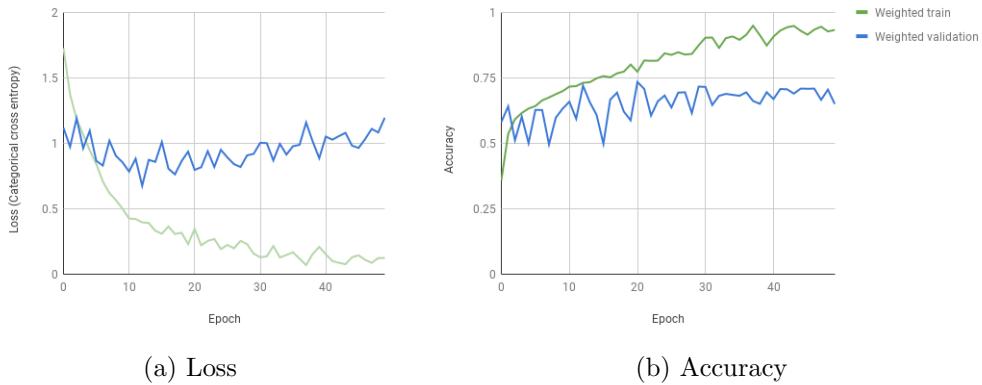


Figure 14: Training and validation loss and accuracy for the transfer learning classification model. The first 25 epochs show the training of the dense layers whereas the second half of the training includes all layers.

metric, the normalized  $\pm 1$  accuracy are  $(0.812, 0.823)$  for the base model and  $(0.781, 0.793)$  for the transfer learning model. Here the base model is better. Since the transfer learning model for classification only brings a very small improvement in the normalized accuracy and worsens the normalized  $\pm 1$  accuracy, we decided to keep the base classification model over the transfer learning model.

The regression model however is considerably better than the base model with all evaluation metrics showing better results. The transfer learning model for regression improves the MSE from 80.9 to 72.8 and the MAE from 4.0 to 0.7. Therefore we decided to keep the new transfer learning model over the base regression model.

	Classification	Regression	
Accuracy	0.65	CRPS	0.0274
Normalized accuracy	0.436	MSE	72.768
Normalized $\pm 1$ accuracy	0.787	MAE	3.119

Table 6: Evaluation metrics for the transfer learning models

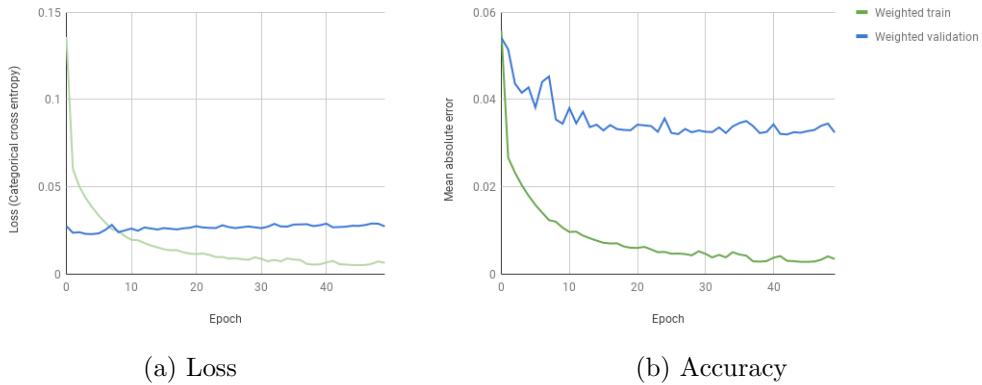


Figure 15: Training and validation loss and accuracy for the transfer learning regression model.

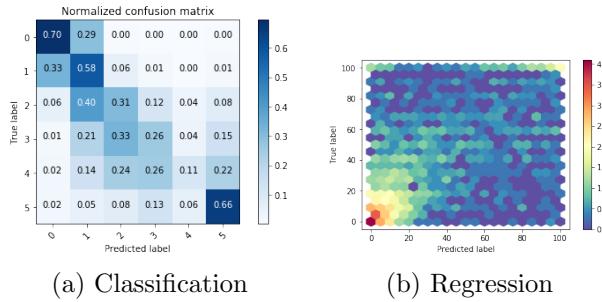


Figure 16: Predictions of the transfer learning models on the validation set.

#### 4.4 Model evaluation

We have now found the best models for classification and regression. Those two models are now validated on the validation set. Figure 17 shows the predictions of the two models for the validation set.

The evaluation metrics for the predictions of the two models for the validation set are shown in ??.

The evaluation metrics of the classification model for the test set are very similar to the ones for the validation set. The classification model performs equally well on the validation and test set. The regression model however performed worse on the test set. Both, the MSE and MAE are higher for the test set than the validation set. This could indicate that the regression

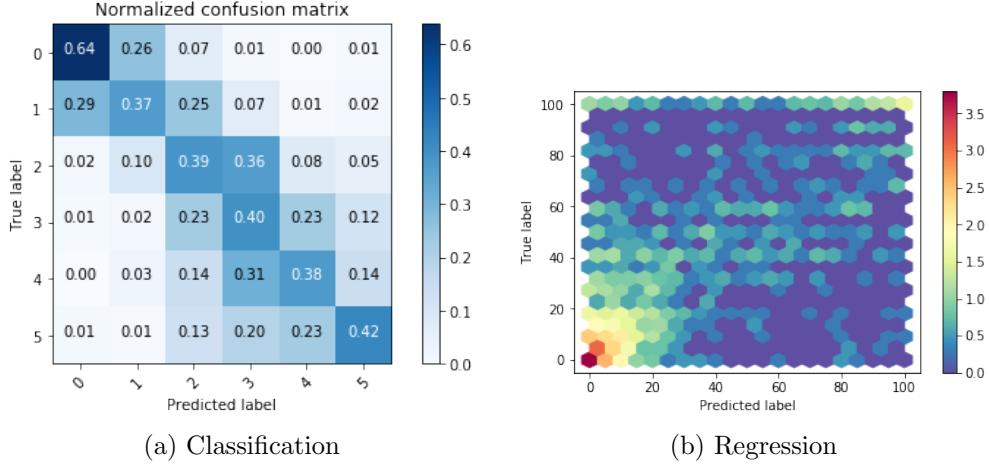


Figure 17: Predictions of the best two models for the test set

Classification		Regression	
Accuracy	0.551	CRPS	0.0309
Normalized accuracy	0.432	MSE	97.586
Normalized $\pm 1$ accuracy	0.832	MAE	3.496

Table 7: Evaluation metrics for the best models evaluated for the validation set

model is overfitting.

#### 4.5 Comparing the classification and regression model

In order to compare the regression model to the classification model the predicted percentages of joint erosion were converted to Ratingen scores. Because the predictions are never exactly zero, we increased the range of class 0 to percentage of bone erosion between 0 and 1. Class 1 is therefore from 1 % to 10 % bone erosion. All other classes are according to subsection 2.2. The resulting confusion matrix is shown in Figure 18 and the corresponding evaluation metrics can be seen in Table 8.

When comparing the confusion matrix for the validation set to c) in Figure 7 and the confusion matrix for the test set to a) in Figure 17, we

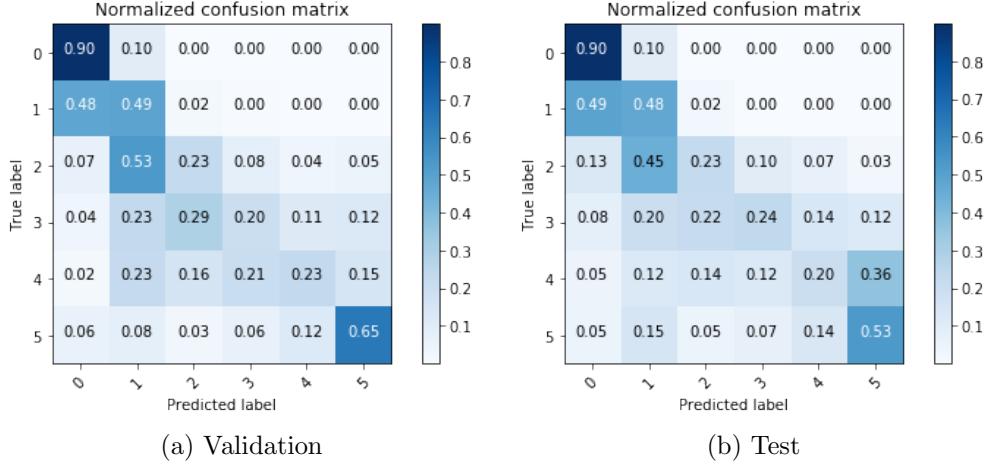


Figure 18: Confusion matrix of predictions of the regression model converted to classes.

can see that the regression model is more accurate for the labels 0, 1 and 5 whereas the accuracy for the labels 2, 3 and 4 is lower. It also seems that are a more outliers compared to the classification model, for example observations with the true label 5 that are mistakenly predicted as label 1.

	Validation	Test
Accuracy	0.762	0.754
Normalized accuracy	0.449	0.429
Normalized $\pm 1$ accuracy	0.799	0.788

Table 8: Evaluation metrics for regression.

The evaluation metrics are compared to c) in Table 4 for the validation set and ?? for the test set. For both the validation and the test set the accuracy of the regression model is higher, since the overrepresented class 1 is predicted more accurately. Whereas the class normalized accuracies are slightly lower than for the classification model.

## 4.6 Attention maps

In order to see where the attention of the models lies, the outputs of intermediate convolutional layers were visualized. Ideally, the last convolutional layer of the model should be visualized. However, the last convolutional layer in our classification model has the dimensions 9 x 9 which is a very low resolution and would lead to a very coarse attention map. Therefore the last convolutional layer with higher dimensions was chosen, its outputs have a dimension of 18 x 18. In the transfer learning regression model, a similar resolution was desired. But the outputs of the last convolutional layer with the dimension 16 x 16 showed attention all over the image. Therefore we had to use a convolutional layer closer to the end of the model. The last convolutional layer with dimension 7 x 7 was chosen, which is very coarse, but shows a centralized attention.

The outputs of every filter of the chosen convolutional layers was added together and normalized to values in the range [0, 1]. It was then colormapped and overlaid onto the original image. Yellow shows high attention whereas purple shows low attention.

Figure 19 shows that the attention of both models focuses on the gap between the two bones most of the time. This is what was desired, since the bone erosion happens in that area.

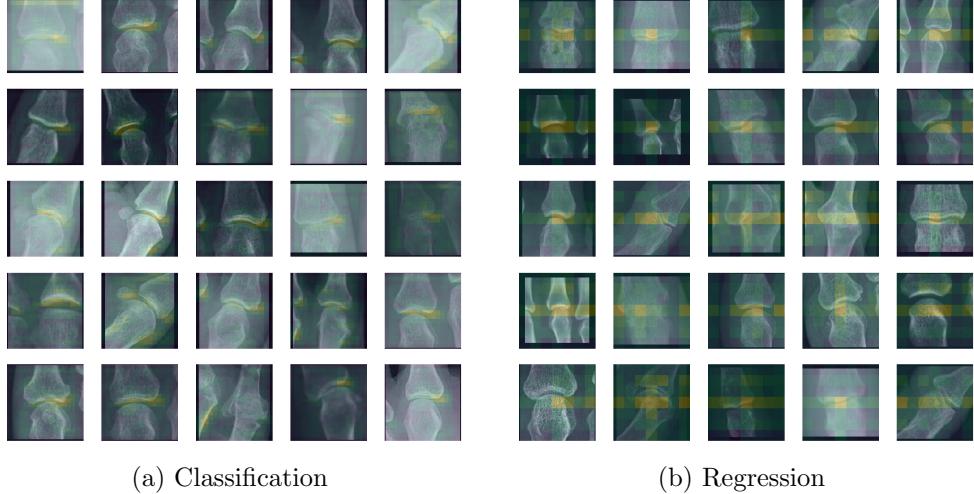
## 4.7 Analysis of the embeddings

To visualize the high-level representations learned by our networks, t-SNE was applied to the outputs of the last hidden layer. Each dot represents an image and the colors represent the true labels of those images. Figure 20 shows that both neural networks managed to separate the different scores quite well. However, there are a few images with low scores in the area of the bad scores.

Visually, it appears like the separation between the different classes is a bit more distinct in the regression model.

### 4.7.1 K-nearest-neighbor of embeddings

By doing a k-nearest-neighbor (KNN) classification on the embeddings of the classification model we can see how well the model separates the different classes. The KNN classification achieved a normalized accuracy of 31 %



(a) Classification

(b) Regression

Figure 19: Attention maps of the best two models for random samples of the validation set

which is worse than the normalized accuracy achieved by the fully connected layers of the CNN of 42 %. And also the normalized  $\pm 1$  accuracy is worse with 63 % compared to 82 %.

????????????? compare to classification transfer learning embeddings

The same can be done with a KNN regression for the embeddings of the regression model. Surprisingly, the KNN regression performs similarly well as the fully connected layers of the CNN. The MAE is a bit worse with 3.7 compared to 3.1, but the MSE is even better with 66.2 compared to 72.8. This means that the

????????????? compare to regression base model embeddings

#### 4.7.2 Analysis of outliers in the embeddings

Next we analyzed the outliers in the embeddings of the regression model. We defined the area on the right side with majoritarian scores higher than 1 as the "bad area" whereas the area on the left side with majoritarian scores equal to zero is called the "good area". Our hypothesis is, that the rheumatologist is more likely to score a damaged joint lower if the other joints of this patient are healthy. The same reasoning can be applied in the reverse. The rheumatologist might be more likely to give a healthy joint a

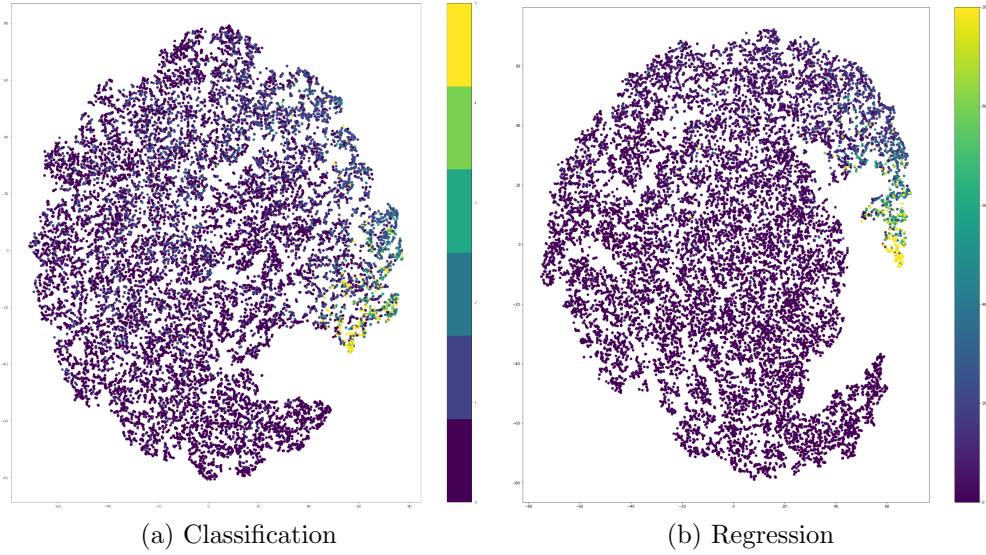


Figure 20: T-SNE of the embeddings. Each point represents an image (joint) the color represents the bone erosion score.

higher score if the other joints of this patient are damaged.

We defined all joints with a score  $\leq 20\%$  (Ratingen-score = 0) as healthy and all joints with a score  $> 20\%$  as damaged. In the "bad area" all healthy joints are seen as outliers whereas in the "good area" all damaged joints are seen as outliers. Figure 21 shows the mean score of patients which have a score that is an outlier compared to all scores that are not outliers.

These findings support our hypothesis. Indeed outliers in the "bad area" seem to be of patients with healthier joints compared to the other patients. The opposite is true for the "good area" as well.

## 4.8 Analysis of correlations between bone erosion and disease activity

In order to determine whether there is a correlation between the bone erosion scores and the disease activity we compared the Rau-score and the DAS. Both scores are described in section 2. It is expected that the disease activity score is correlated with the bone erosion score. However, Figure 22 shows that there is only a very weak correlation. Between the two different DAS

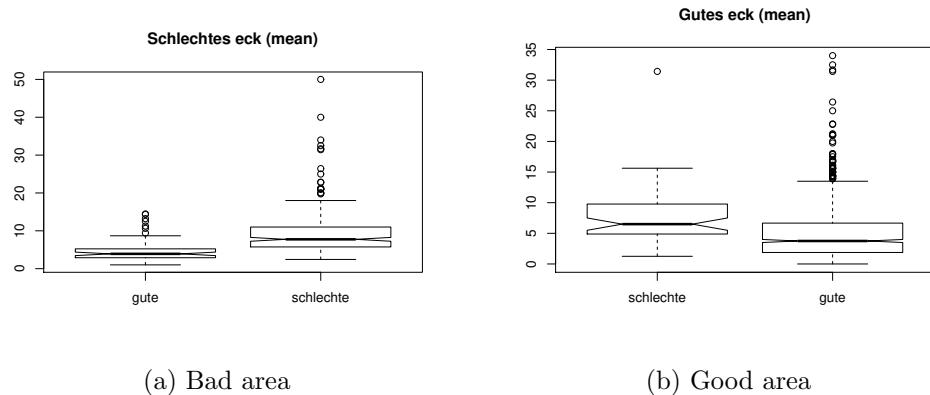


Figure 21: Average score of patients which have a score that is an outlier.

and the Rau-score.

This surprising result suggests that the amount of bone erosion is only a minor factor in the disease activity.

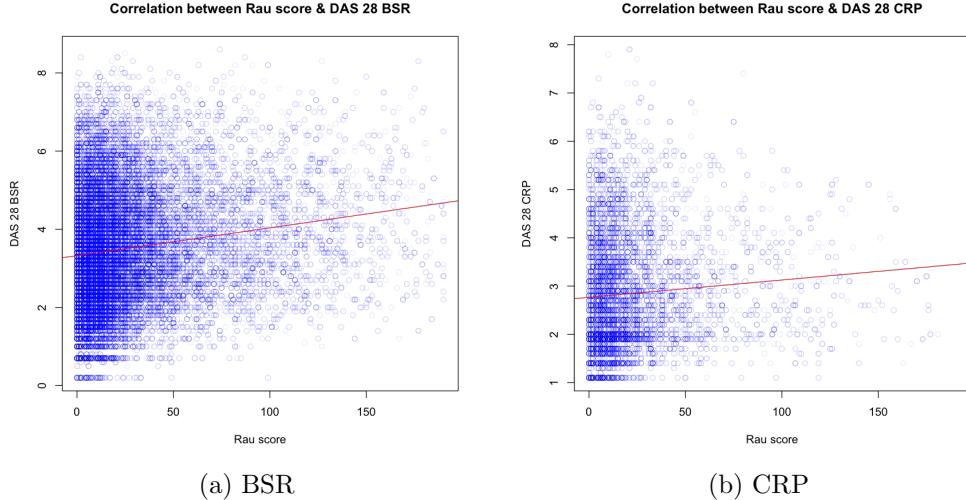


Figure 22: Scatter plot of DAS compared to Rau-scores

## 5 Discussion

In the previous sections two good models were found. One for classification and one for regression. When transforming the predictions of the regression model into classes, the normalized accuracies of the two models are almost equal. However, the predictions are very differently distributed. The classification model performs similarly well for all classes whereas the regression model performs clearly better for the extreme classes 0, 1 and 5.

It is difficult to choose one model over the other. The classification model would be better suited if a better accuracy for the intermediate classes 2, 3 and 4 is desired, whereas the regression model is better suited if accurate predictions of the extreme classes 0, 1 and 5 is desired.

An analysis of the outliers in the T-SNE shows, that there might have been a bias when the images were scored. It seems that if a patient has many damaged joints, a healthy joint is more likely to get a worse score. The opposite is true as well.

In the last section we found, that there is only a minor correlation between the percentage of bone erosion and the disease activity. While the bone erosion measures the physical damage to the joint, the disease activity score measures how much the patient suffers from the disease. Given bone ero-

sion scores, it is almost impossible to draw conclusions regarding the disease activity.

## 6 Conclusion and outlook

The predictions of the classification model give a good estimate for the Ratingen-score and the predictions of the regression model give a good estimate for the percentage of bone erosion. But the models are not as good as a trained professional. Many predictions are off by one Ratingen class, which corresponds to 20 % of bone erosion. Occasional outliers are as far off as 100 %, meaning that completely eroded joints are mistakenly classified as healthy joints or reverse. It is unclear whether an automation of the scoring process with these models can save time, since the rheumatologist still has to take a second look at the results in order to find misclassifications.

Further research could examine whether the certainty for predictions is a good indicator for outliers. In that case, the rheumatologist would only have to look at cases, where the model wasn't certain.

The attention maps showed us that the model seemed to focus on the desired part of the image. A more sophisticated approach to detect which parts of the image are responsible for the predictions, would be occluding parts of the image. A black rectangle is slided over the image do occlude different parts. For every position of the black rectangle a probability for the correct prediction can be calculated and shown as a heatmap.

Since the bone erosion scores and the disease activity are only slightly correlated, it would be interesting to examine whether the disease activity could be predicted from the images. For this task, only a slight modification to our models is necessary. One could use the classification model and change its input layer depth to 10. Instead of color channels, every channel would hold an image of a different joint from one hand.

In addition it should be investigated how different loss functions change the distribution of the predictions. For example the output layer of the regression model could be changed to only one node and the CRPS loss function could be replaced with the MSE loss function.

## References

- [1] American College of Rheumatology. *Rheumatoid Arthritis*. [Online; accessed 26-September-2017]. URL: <https://www.rheumatology.org/I-Am-A/Patient-Caregiver/Diseases-Conditions/Rheumatoid-Arthritis>.
- [2] SCQM. *About Us*. [Online; accessed 11-October-2017]. URL: <https://www.scqm.ch/en/ueber-uns/>.
- [3] Seantis GmbH. *Data Driven Web Applications*. [Online; accessed 12-October-2017]. URL: <https://www.seantis.ch>.
- [4] Nima Tajbakhsh et al. “Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?” In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1299–1312. URL: <https://doi.org/10.1109/TMI.2016.2535302>.
- [5] Rahul Paul et al. *Make Your Bone Great Again : A study on Osteoporosis Classification*. University of South Florida, 2017. URL: <https://arxiv.org/abs/1707.05385>.
- [6] Zhi-Hua Zhou et al. “Lung cancer cell identification based on artificial neural network ensembles”. In: *Artificial Intelligence in Medicine* 24.1 (2002), pp. 25–36. URL: [https://doi.org/10.1016/S0933-3657\(01\)00094-X](https://doi.org/10.1016/S0933-3657(01)00094-X).
- [7] Matthew Chen. *Automated Bone Age Classification with Deep Neural Networks*. Stanford University, 2016. URL: [http://cs231n.stanford.edu/reports/2016/pdfs/310\\_Report.pdf](http://cs231n.stanford.edu/reports/2016/pdfs/310_Report.pdf).
- [8] Paulina Hensman and David Masko. *The Impact of Imbalanced Training Data for Convolutional Neural Networks*. KTH Royal Institute of Technology, 2015. URL: [https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko\\_dkand15.pdf](https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf).
- [9] R. Rau and S. Wassenberg. “Scoringmethoden bei der rheumatoiden Arthritis”. In: *Bildgebende Verfahren in der Rheumatologie*. Ed. by Deutsche Gesellschaft für Rheumatologie. Steinkopff, 2007. Chap. 2, pp. 27–46. URL: [https://doi.org/10.1007/978-3-7985-1721-9\\_2](https://doi.org/10.1007/978-3-7985-1721-9_2).

- [10] Radboud University Nijmegen Medical Centre (RUNMC). *Twenty-eight joints*. [Online; accessed 12-October-2017]. URL: <https://www.das-score.nl/das28/en/difference-between-the-das-and-das28/how-to-measure-the-das28/twenty-eight-joints.html>.
- [11] Radboud University Nijmegen Medical Centre (RUNMC). *Alternative validated formulae*. [Online; accessed 12-October-2017]. URL: <https://www.das28.nl/das28/en/difference-between-the-das-and-das28/how-to-measure-the-das28/how-to-calculate-the-das28/alternative-validated-formulae.html>.
- [12] Stanford University Andrej Karpathy. *CS231n Convolutional Neural Networks for Visual Recognition*. [Online; accessed 26-September-2017]. URL: <https://cs231n.github.io/>.
- [13] R. Kruse et al. *Computational Intelligence*. Springer, London, 2016. URL: [https://doi.org/10.1007/978-1-4471-7296-3\\_2](https://doi.org/10.1007/978-1-4471-7296-3_2).
- [14] *Python Language Reference, version 3.5*. Python Software Foundation. URL: <https://www.python.org/>.
- [15] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [16] François Chollet et al. *Keras*. 2015. URL: <https://github.com/fchollet/keras>.
- [17] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [18] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2017. URL: <https://www.R-project.org/>.
- [20] Kaggle. *Second Annual Data Science Bowl*. [Online; accessed 26-September-2017]. URL: <https://www.kaggle.com/c/second-annual-data-science-bowl#evaluation>.
- [21] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. URL: <https://arxiv.org/abs/1512.00567>.

## List of Figures

1	Proximal interphalangeal (PIP) joints and carpometacarpal (MCP) joints of the left hand. Cropped images of these ten joints were used to train the neural networks. . . . .	9
2	Feed forward neural network with one hidden layer. . . . .	11
3	Structure of a convolutional neural network. . . . .	12
4	Histogram of the bone erosion scores . . . . .	14
5	Architecture of the CNN for classification . . . . .	17
6	Training and validation loss and accuracy for the three classification models. . . . .	18
7	Normalized confusion matrices for the predictions of the classification models on the validation set. . . . .	18
8	Architecture of the CNN for regression . . . . .	20
9	CRPS probability score . . . . .	21
10	Example of a prediction . . . . .	22
11	Training and validation loss and mean absolute error for the three regression models. . . . .	22
12	Predictions of the regression models on the validation set. . . . .	22
13	Model architecture for transfer learning . . . . .	23
14	Training and validation loss and accuracy for the transfer learning classification model. The first 25 epochs show the training of the dense layers whereas the second half of the training includes all layers. . . . .	24
15	Training and validation loss and accuracy for the transfer learning regression model. . . . .	25
16	Predictions of the transfer learning models on the validation set. . . . .	25
17	Predictions of the best two models for the test set . . . . .	26
18	Confusion matrix of predictions of the regression model converted to classes. . . . .	27
19	Attention maps of the best two models for random samples of the validation set . . . . .	29
20	T-SNE of the embeddings. Each point represents an image (joint) the color represents the bone erosion score. . . . .	30
21	Average score of patients which have a score that is an outlier. . . . .	31
22	Scatter plot of DAS compared to Rau-scores . . . . .	32

## List of Tables

1	Disease stages of the Ratingen-score [9]	8
2	Columns of main dataset	13
3	Columns of secondary dataset	13
4	Evaluation metrics for classification.	19
5	Evaluation metrics for regression.	21
6	Evaluation metrics for the transfer learning models	24
7	Evaluation metrics for the best models evaluated for the validation set	26
8	Evaluation metrics for regression.	27