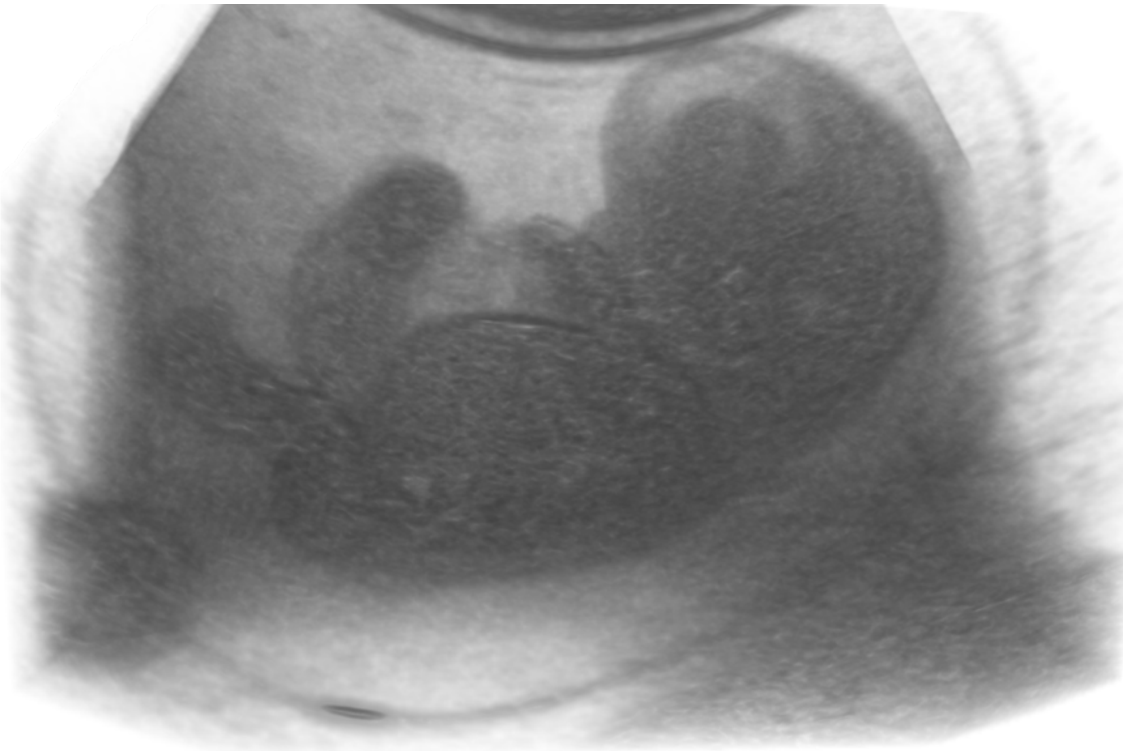
Automated Carnegie Staging of the Human Embryo in 3D Ultrasound using Deep Learning

Ruben Niemantsverdriet

February 2024



Automated Carnegie Staging of the Human Embryo in 3D Ultrasound using Deep Learning

By

Ruben Niemantsverdriet

in partial fulfillment of the requirements for the degree of

Master of Science
in Biomedical Engineering
Track Medical Physics

at the Delft University of Technology,
to be defended publicly on Tuesday February 27, 2024 at 14:00 PM.

Thesis committee:	Dr. F.M. Vos, chair	TU Delft
	Ir. W.A.P. Bastiaansen	Erasmus MC
	Dr. ir. S. Klein	Erasmus MC
	Dr. M. Rousian	Erasmus MC
	Dr. A.H.J. Koning	Erasmus MC
	Dr. J.F. Veenland	Erasmus MC

Abstract

The periconceptional period, encompassing the embryonic phase, is a critical window where a majority of reproductive failures, pregnancy complications, and adverse pregnancy outcomes arise. The Carnegie staging system comprises 23 stages which are based on embryonic morphological development. This allows for the assessment of normal and abnormal embryonic development during this critical period. In-utero Carnegie staging using three-dimensional (3D) ultrasound scans visualized with virtual reality offers valuable insights but is currently a time-consuming manual process. To address this, we propose a deep learning approach for Carnegie staging in 3D ultrasound scans.

We used a dataset comprising 1413 3D ultrasound scans from the Rotterdam Periconceptional Cohort, annotated with Carnegie stages spanning from stages 13 to 23, including fetal subjects. Various training strategies were explored. We compared a metric regression approach, which considers the ordered nature of the Carnegie stages by treating the Carnegie stages as a continuous variable, with a multi-class classification approach, treating stages as independent categories. Additionally, we evaluated the influence of using a loss function accommodating the categorical nature of the Carnegie stages in the metric regression approach and examined the impact of incorporating embryonic size in the model input. Ultimately, a regression approach using the Mean Squared Error (MSE) loss function emerged as the optimal choice.

This model achieved a classification accuracy of 0.59 and a Root Mean Squared Error (RMSE) of 0.62 on the test set. This performance is comparable to an intermediate human rater, which achieved an accuracy of 0.63 and a RMSE of 0.65. Our findings represent a significant step towards the development of an automated Carnegie staging method, offering the potential for a more comprehensive evaluation of the critical embryonic phase in the clinic.

1 Introduction

The pregnancy is divided into an embryonic and a fetal period. During the embryonic period, which consists of the first 8 weeks of pregnancy, many features, including most of the organs in the human body, start to develop rapidly [1]. The embryonic period is part of the periconceptional period, which spans from 14 weeks before conception to 10 weeks after conception. The majority of reproductive failures, pregnancy complications, and adverse pregnancy outcomes stem from this period [2].

The Carnegie staging system facilitates the assessment of normal embryonic development in terms of morphology during this critical period [3]. Additionally, it enables the identification of abnormalities by comparing the embryonic size or age with those associated to the corresponding Carnegie stage [4]. The system comprises 23 stages that describe the entire embryonic period, starting at a fertilized oocyte and ending with an embryo in which all essential internal organ systems are present. Although the Carnegie stages are correlated with embryonic size and age, they are based only on the internal and external morphological characteristics of the embryo [5]. Examples of each of the Carnegie stages are shown in Figure 1. Additionally, 3D renderings of segmented embryos in 3D ultrasound are shown in Figure 2.

3D ultrasound scans made during the embryonic period allow for in-utero Carnegie staging. Currently, manual rating of these scans involves the use of a virtual reality system, as described by Rousian et al. [7], which enables depth perception. However, this approach is labor-intensive and time-consuming. To enable the use of Carnegie staging on a larger scale without relying on an elaborate virtual reality system, there is a need for the development of an automated Carnegie staging method.

To address this challenge, we aim to develop a deep learning method that predicts the Carnegie stage

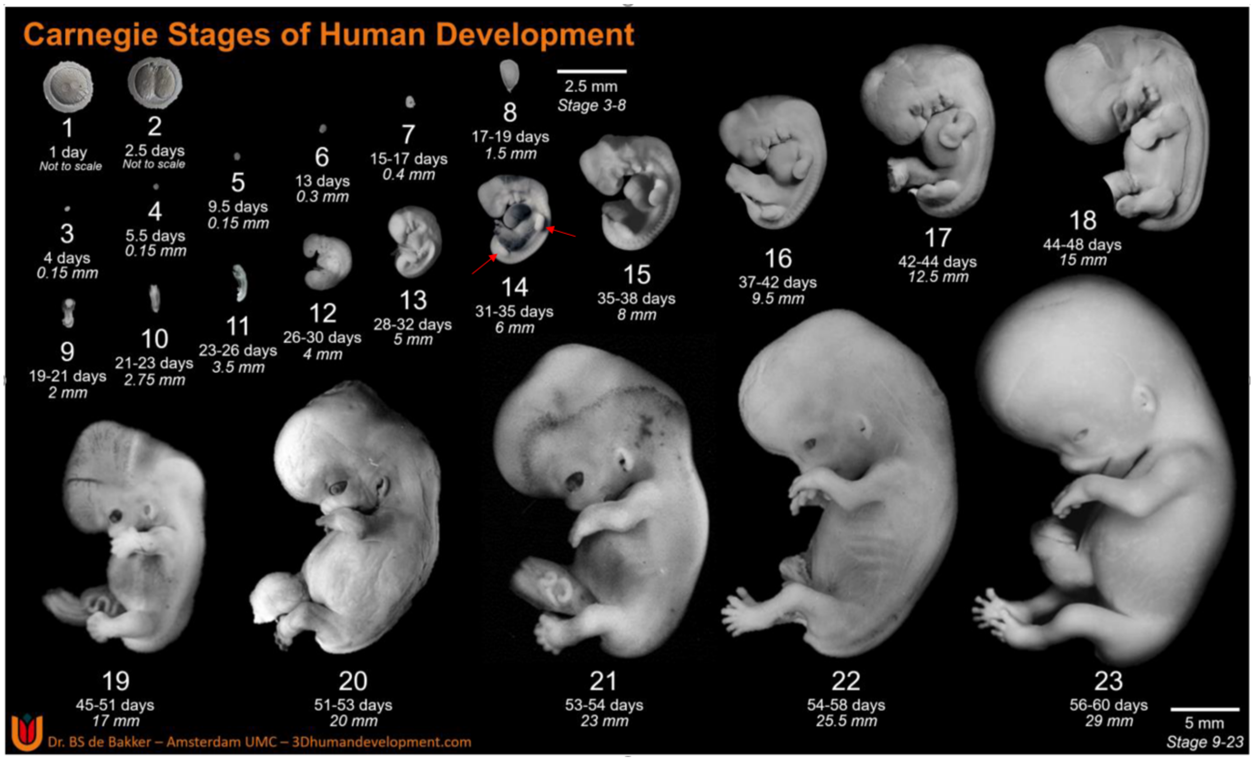


Figure 1: Illustrative images showcasing embryos at all 23 Carnegie stages obtained from Flierman et al. [6]. Microscope images were used for stages 1 and 2, 3D reconstructions based on histological sections for stages 3-8, and photos from subjects in the Carnegie collection for stages 9-23. Additionally, the mean embryonic age and length for each stage are provided. Red arrows indicate the upper and lower limbs in stage 14.

based on the 3D ultrasound scan. We opted for a deep convolutional neural network (CNN) due to its established effectiveness in medical imaging [8], using an architecture inspired by the Densely Connected Convolutional Network (DenseNet) by Huang et al. [9].

We hypothesized that a metric regression approach, mentioned by Niu et al. [10], which considers the ordered nature of Carnegie stages, would outperform a multi-class classification approach treating the classes as independent. To test this, both methods were implemented. In the metric regression approach, the Carnegie stages, although categorical, were treated as a continuous variable, with network output rounded to the nearest stage for accuracy calculation. We further hypothesized that employing a loss function accommodating this rounding would enhance model accuracy. Thus, multiple networks were trained under identical conditions but with varying loss functions to assess this hypothesis.

Efforts were undertaken to prevent the model from accessing information regarding embryonic size, as it should rely solely on morphological features for predicting Carnegie stages. Nonetheless, the inclusion of embryonic size information could potentially enhance the model's performance. To explore this possibility, we examined the impact of incorporating embryonic size into the model input on its overall performance.

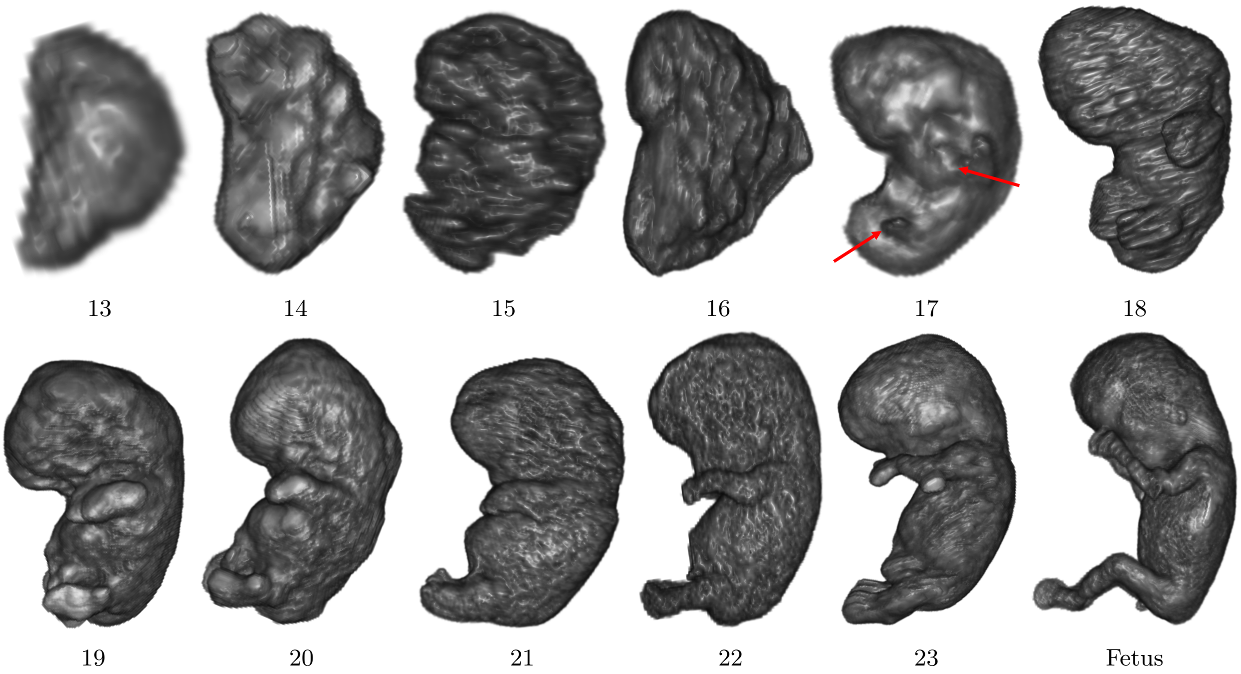


Figure 2: Examples of 3D renderings of 3D ultrasound images of different not-to-scale embryos of the Carnegie stages present in the dataset. The embryonic volumes were segmented and used to mask the 3D ultrasound scan. The red arrows indicate the upper and lower limbs in stage 17.

A literature review was conducted to provide an overview of relevant literature about Carnegie staging in 3D ultrasound, regularization techniques, and datasets with ordered labels and is included as the appendix of this thesis.

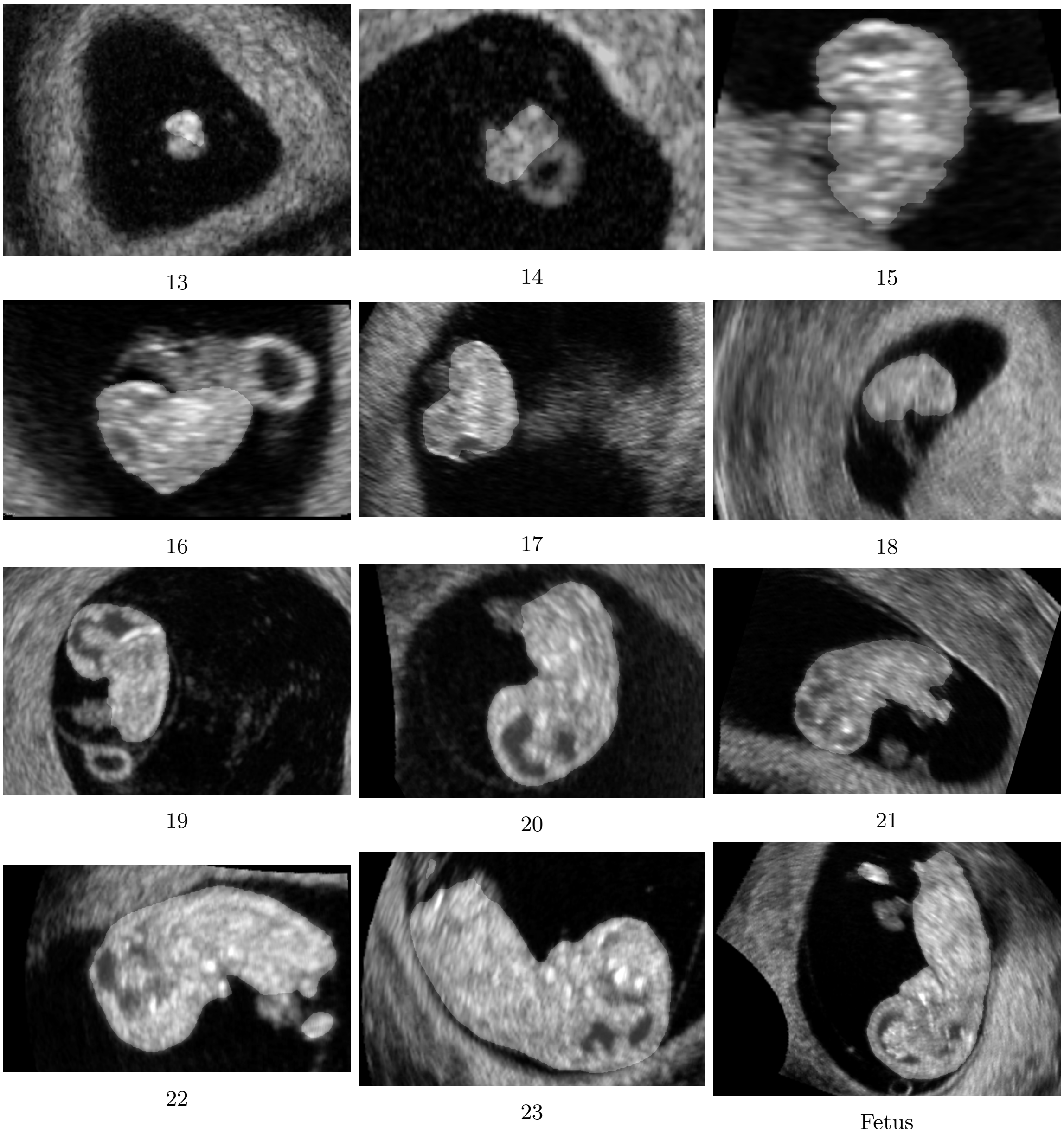
To the best of our knowledge, this is the first attempt to develop a deep learning-based Carnegie staging method, which could pave the way for the broader application of Carnegie staging in research and clinical practice.

2 Data

2.1 Dataset

The dataset for this project was obtained from the Rotterdam Periconceptional Cohort from the Erasmus MC, University Medical Center Rotterdam, the Netherlands. This ongoing cohort study investigates the effects of maternal and paternal health during the periconceptional period of pregnancy [11]. In this cohort study, ultrasound examinations were performed multiple times during the pregnancy. During the embryonic period, transvaginal 3D ultrasound scans were acquired between 6 and 12 weeks of gestational age (GA). These scans were made using a handheld 4.5 - 11.9 MHz vaginal probe of a Voluson E8 or E10 (GE Healthcare, Austria) [12]. All scans had isotropic voxels, however, the voxel size, number of voxels in the scans, and field-of-view of the scans varied. Because of the natural variation in the location and orientation of the embryo in the uterus and the variation in scan acquisition, the orientation of the embryo in the scan also varied. Figure 3 displays examples of all Carnegie stages present in the dataset, illustrating the variability in ultrasound scans and embryo orientation.

The Carnegie stage was known for 1609 of these



13

14

15

16

17

18

19

20

21

22

23

Fetus

Figure 3: Examples of mid-sagittal planes of 3D ultrasound images of embryos with different Carnegie stages. The segmentation of the embryonic volume is displayed with a brighter hue.

Figure 4: Picture of the BARCO I-space virtual reality system retrieved from Verwoerd-Dikkeboom et al. [3]. A 3D ultrasound scan of an embryo is displayed with the V-scope volume rendering application.

ultrasound examinations. The Carnegie staging was performed by two trained human observers. This was done by visually inspecting the 3D ultrasound scans in the BARCO I-space virtual reality system [13]. Here, the observer was surrounded by stereo images that allow for depth perception and 3D interaction with the data, as described in more detail by Verwoerd-Dikkeboom et al. [3, 14]. The V-Scope volume rendering application was used to generate a dynamic holographic representation of the embryo, allowing manipulation and measurement in three dimensions. A picture of the I-space can be seen in Figure 4.

The Carnegie staging was done according to the Carnegie stage criteria by O'Rahilly and Müller [1, 15], using only morphological characteristics of the embryo. The limb development and curvature of the trunk were mostly used as external characteristics, while the brain ventricle development was used as internal characteristic [12, 13]. Embryonic staging generally took 1-2 minutes per embryo [13]. Next to the Carnegie stage, embryonic volume (EV), crown-rump length (CRL), and GA of the embryo were known for 94% of the scans in the dataset.

The EV and CRL were measured semi-automatically in the I-space virtual reality system by trained human observers [12]. For naturally conceived pregnancies (63% of scans), GA was determined from the first day of the last menstrual period (LMP), with adjustments for irregular cycles outside the 25 to 31-day range [16]. When there was a discrepancy of over 7 days between LMP-derived GA and CRL-based GA, or in the absence of LMP data, GA was estimated using the CRL from the 9-week ultrasound. For in-vitro fertilization (IVF) or intracytoplasmic sperm injection (ICSI) pregnancies (37% of scans), GA was calculated by adding 14 days to the date of oocyte retrieval. In cases of cryopreserved embryo use (12% of scans), GA was determined by adding 19 days to the transfer date.

2.2 Preprocessing

The embryonic volumes were segmented using the method by Bastiaansen et al. [17], which is based on the nnU-Net proposed by Isensee et al. [18]. We only included scans for which the relative volume error of the automatically segmented embryo compared to the measured EV was below 0.3. In case multiple scans per ultrasound examination were available, we included the scan with the lowest relative volume error.

This resulted in a dataset of 1413 3D ultrasound scans of embryos with known Carnegie stage, GA, EV, and CRL. A histogram of the number of scans available per Carnegie stage is shown in Figure 5. The Carnegie stage labels ranged from stage 13 to stage 23. Additionally, some embryos had surpassed the embryonic stage and were labeled as fetuses. Ultrasound examinations were performed at multiple time points during the pregnancy. This resulted in the number of included scans per subject at different time points ranging from one to three. However, for most subjects, only one scan per subject is available.

The preprocessing steps for two example scans are shown in Figure 6. The segmented embryonic volumes were used to mask the original scans. The masked scan was cropped to contain only the segmentation, with an extra margin of five voxels to each side, see Figure 6c and 6g. The number of voxels in the cropped scans varied from 21 to 257 in the first

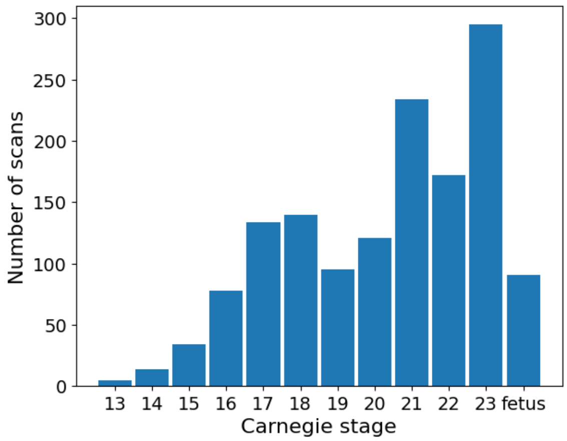


Figure 5: Histogram of the number of labeled scans available per Carnegie stage. The scans labeled as a fetus are also included.

dimension, 19 to 211 in the second dimension, and 24 to 231 in the third dimension. The two smallest dimensions of the masked scans were symmetrically padded with zeros until they were the same size as the largest dimension of the scan. All scans were isotropically resized to 200x200x200 voxels using trilinear interpolation, see Figure 6d and 6h. Lastly, all image intensities were normalized to be between 0 and 1.

These preprocessing steps were implemented to mitigate the influence of embryonic size on the network input. By removing the surroundings of the embryo from the input and ensuring that the number of voxels comprising the embryo is unrelated to its size, efforts were made to prevent the network from inferring embryonic size and incorporating this information into the Carnegie stage prediction process.

2.3 Dataset split

The dataset was split into a training, validation, and test set, containing 60%, 20%, and 20% of the data respectively. This split was performed on the subject level to ensure that the validation and test set did not include scans of subjects in the training set. Additionally, it was ensured that the Carnegie stage and GA distributions in all three sets were similar so that all sets are representative and to ensure a reliable evaluation of the model performance. This resulted in a training, validation, and test set of 845, 296, and 272 samples respectively.

2.4 Data augmentation

To address the issue of class imbalance, as is visible in Figure 5, label-stratification was performed by oversampling the under-represented classes in the training set. This was done by uniformly drawing 300 random samples from each class in the training set. All samples were then randomly rotated by a multiple of 90 degrees in all three directions. This was done with equal probability for either 0, 90, 180, or 270 degrees rotation, resulting in $4^3 = 64$ possibilities. This type of data augmentation was chosen because of the existing variation in embryo orientation in the input scan. Only multiples of 90-degree rotations were used to prevent part of the segmented embryo from ending up outside the cropped scan. This resulted in an augmented training set of $12 \cdot 300 = 3600$ samples.

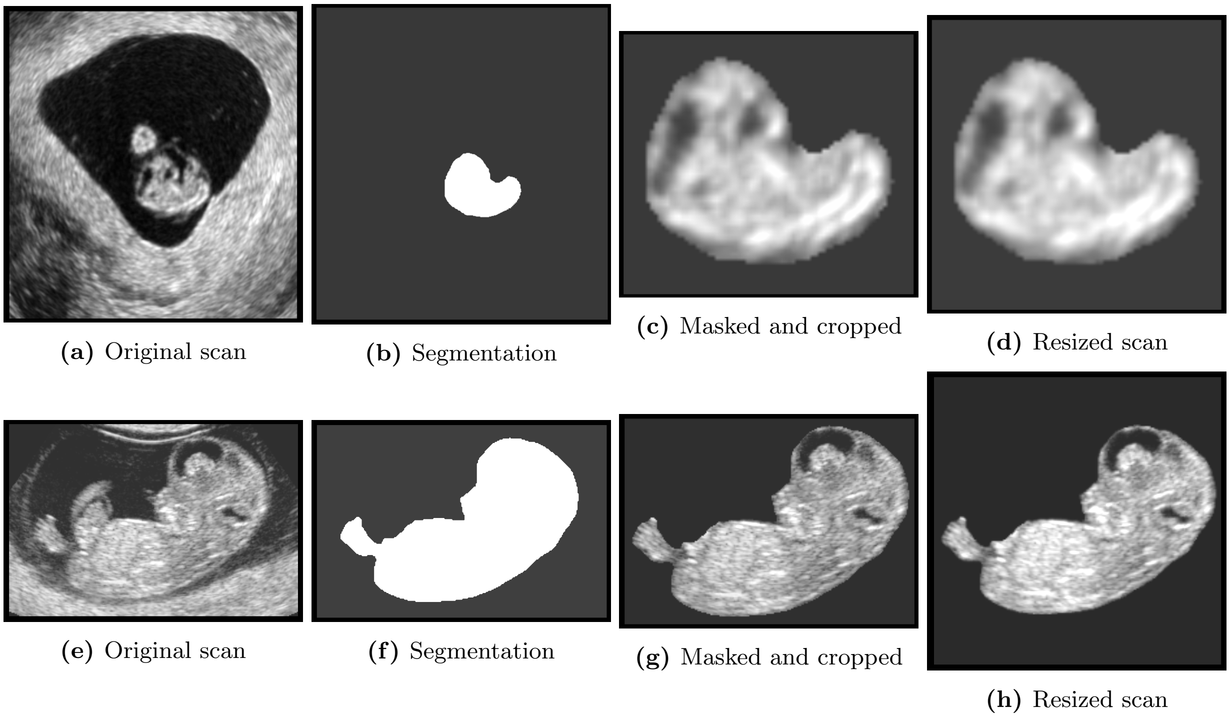
3 Methods

3.1 Problem definition

We are given a set of 3D input images x with ground truth Carnegie stage labels y. y ranges from Carnegie stage 13-23, including fetal subjects, so that the number of classes in the dataset is $C = 12$. The goal is to train the network $f_\theta : x \to \hat{y}$, with parameters θ and output Carnegie stage $\hat{y}$, so that $\hat{y} = y$ for as many x in the validation or test set.

3.2 Network architecture

We used the 3D DenseNet-121 from the Medical Open Network for Artificial Intelligence (MONAI) version 1.2.0 (code available at `https://github.com/Project-MONAI`). DenseNets employ direct concatenation of feature maps between layers of matching sizes to ensure that each layer has direct access to gradients from the loss function and the original input signal, leading to improved training efficiency [9].


(a) Original scan

(b) Segmentation

(c) Masked and cropped

(d) Resized scan

(e) Original scan

(f) Segmentation

(g) Masked and cropped

(h) Resized scan

Figure 6: Example of preprocessing steps for an embryo of Carnegie stage 17 (**a-d**) and Carnegie stage 22 (**e-h**). Only a sagittal slice of the 3D ultrasound scan is displayed.

Additionally, DenseNets require fewer parameters for training due to their efficient feature propagation and reuse, resulting in a regularizing effect [19].

A general overview of the DenseNet-121 architecture can be seen in Figure 7. The input image is shown on the left. First, convolution (C1) and max pooling (P1) operations are performed [9]. These are followed by dense blocks (Dx) and transition layers (Tx). Each dense block consists of multiple dense layers, namely 6, 12, 24, and 16 dense layers for D1, D2, D3, and D4 respectively. Each dense layer consists of a batch norm (BN), a rectified linear unit (ReLU), a 1x1x1 convolution, a BN, a ReLU, and a 3x3x3 convolution, in this order. Each dense layer performs these operations on the input feature map and concatenates the result to the feature map. This increases the number of feature maps in the feature space by 32 for each dense layer. The transition layers apply a 1x1x1 convolution and max-pooling operation, reducing feature map dimensionality by one-half. After the last dense block, a global average pooling is performed to form the final feature map, which is connected to a fully connected layer (FC).

Three approaches for the network output were used:

1. The metric regression approach, shown on the top in Figure 7, used a single output channel that directly gives the Carnegie stage prediction as a continuous variable. This was done by connecting the last fully connected layer directly to the output channel and using the Carnegie stages directly as labels.

2. In the multi-class classification approach, a Softmax function was first applied and the number of output channels was the same as C. This is displayed as the middle output approach in Figure

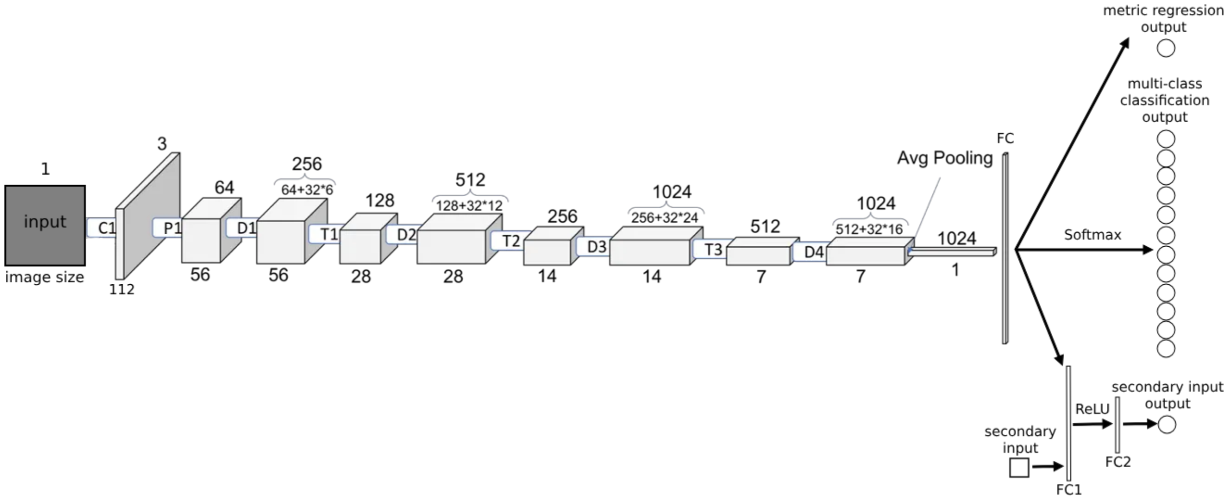


Figure 7: Schematic overview of DenseNet-121 with different output structures. The rectangular blocks represent the feature spaces. The value below the feature spaces represents both the height, width, and depth of the feature space and the value above represents the number of feature maps in the feature space, along with the derivation. The arrows indicate connections between layers. Image adapted from [20].

   7. The Carnegie stage labels were transformed into one-hot encoded vectors.

3. A third network architecture was used, displayed at the bottom of Figure 7, that allows for secondary input variables, such as EV or the voxel size. This was done by adding two fully connected layers, FC1 and FC2 with layer widths W_{FC1} and W_{FC2}, after the last layer of the DenseNet. The secondary input variable was then concatenated to the output of the DenseNet to create the input to the extra layers. A ReLU function was added after the first extra fully connected layer to allow non-linear relations in the network. The network output is calculated from FC2 in the same manner as in the metric regression approach.

   This approach was inspired by Wang et al. [21]. Here, a 3D regression CNN was implemented to predict the brain age, where 3D gray matter brain images were used as input. In addition, the sex of the subject was provided as a secondary input to allow the network to adjust for gray matter differences between males and females.

3.3 Loss functions

Different loss functions were used during training. These are described separately below. The loss $\mathcal{L}(\hat{y}, y)$ against the prediction error for a single case $|\hat{y}_n - y_n|$ is plotted in Figure 8a for the different loss functions.

3.3.1 Metric regression

The following loss functions were used in the metric regression approach:

- Mean Squared Error (MSE) loss $\mathcal{L}_{\mathrm{MSE}}(\hat{y}, y)$: The MSE loss function is given in Equation 1, where N represents the number of predictions. In Figure 8a, it can be seen that the gradient of $\mathcal{L}_{\mathrm{MSE}}(\hat{y}, y)$ keeps increasing as the error increases, thereby penalizing outliers more severely.

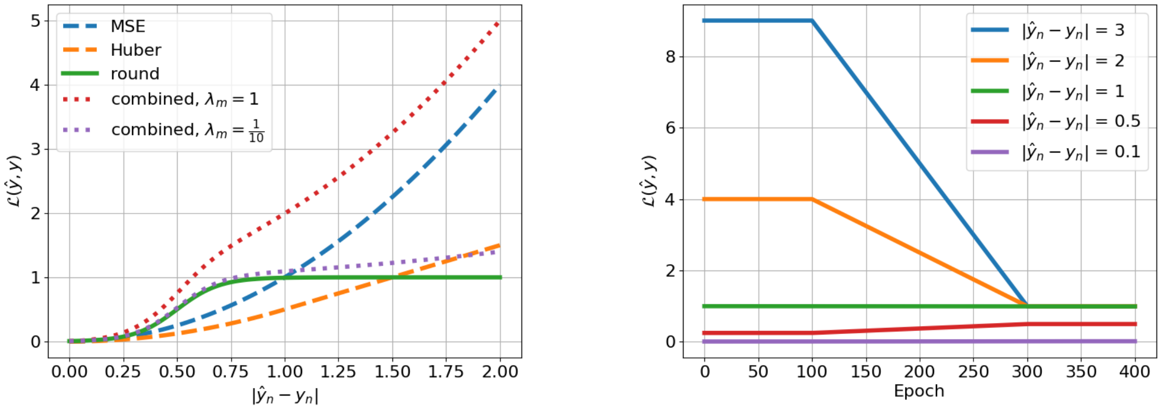
- Huber loss $\mathcal{L}_{\mathrm{Huber}}(\hat{y}, y)$: The Huber loss function is given in Equation 2, where δ denotes the transition point between a squared loss and a linear loss. This way, $\mathcal{L}_{\mathrm{Huber}}(\hat{y}, y)$ is similar to

$$\mathcal{L}_{\mathrm{MSE}}(\hat{y}, y) = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2 \tag{1}$$

$$\mathcal{L}_{\mathrm{Huber}}(\hat{y}, y) = \frac{1}{N}\sum_{n=1}^{N}\begin{cases}\frac{1}{2}(\hat{y}_n - y_n)^2 & \text{if } |\hat{y}_n - y_n| < \delta \\ \delta\left(|\hat{y}_n - y_n| - \frac{\delta}{2}\right) & \text{otherwise}\end{cases} \tag{2}$$

$$\mathcal{L}_{\mathrm{round}}(\hat{y}, y) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2} + \frac{1}{2}\tanh\left(k\left(|\hat{y}_n - y_n| - \frac{1}{2}\right)\right)\right) \tag{3}$$

$$\mathcal{L}_{\mathrm{CE}}(\hat{y}, y) = \frac{1}{N}\sum_{n=1}^{N}\left(-\sum_{c=1}^{C}\log\left(\frac{\exp(\hat{y}_{n,c})}{\sum_{i=1}^{C}\exp(\hat{y}_{n,i})}y_{n,c}\right)\right) \tag{4}$$



(a) Loss against the prediction error for different loss functions. $\delta = 1$ for $\mathcal{L}_{\mathrm{Huber}}(\hat{y}, y)$ and $k = 5$ for $\mathcal{L}_{\mathrm{round}}(\hat{y}, y)$.

(b) Loss over the training epochs for $\mathcal{L}_{\mathrm{shift}}(\hat{y}, y)$ with different prediction errors. $k = 5$ for $\mathcal{L}_{\mathrm{round}}(\hat{y}, y)$.

Figure 8: Plots of loss functions used in the regression approach.

$\mathcal{L}_{\text{MSE}}(\hat{y}, y)$ whenever $|\hat{y}_n - y_n| < \delta$, but is a linear loss scaled with δ otherwise. This causes this loss to penalize prediction errors larger than δ less than $\mathcal{L}_{\text{MSE}}(\hat{y}, y)$.

3.3.2 Rounding loss

The rounding loss function, $\mathcal{L}_{\text{round}}(\hat{y}, y)$, is described in Equation 3, where k controls the steepness of the loss. As can be seen in Figure 8a, $\mathcal{L}_{\text{round}}(\hat{y}, y)$ has a maximum gradient at $|\hat{y}_n - y_n| = \frac{1}{2}$, which is the point up till where a rounded prediction yields a correct result. The goal of using $\mathcal{L}_{\text{round}}(\hat{y}, y)$ was that the network optimization specifically penalizes cases that have a prediction error of just above $\frac{1}{2}$, causing these scans to be misclassified after rounding.

The gradient of $\mathcal{L}_{\text{round}}(\hat{y}, y)$ vanishes when $|\hat{y}_n - y_n|$ becomes much larger than $\frac{1}{2}$, potentially making training with stochastic gradient descent difficult. To overcome this, combined loss functions were used. These are defined as $\mathcal{L}_{\text{combined}}(\hat{y}, y) = \lambda_r \mathcal{L}_{\text{round}}(\hat{y}, y) + \lambda_m \mathcal{L}_{\text{MSE}}(\hat{y}, y)$, where λ_r and λ_m are weight factors. $\mathcal{L}_{\text{combined}}(\hat{y}, y)$ is plotted in Figure 8a with $\lambda_r = 1$, $\lambda_m = 1$ and $\lambda_r = 1$, $\lambda_m = \frac{1}{10}$. Here it can be seen that $\mathcal{L}_{\text{combined}}(\hat{y}, y)$ yields an increased gradient magnitude at $|\hat{y}_n - y_n| = \frac{1}{2}$, without having the issue of having a very small gradient magnitude when $|\hat{y}_n - y_n|$ becomes much larger than $\frac{1}{2}$.

3.3.3 Shifting loss

Using $\mathcal{L}_{\text{round}}(\hat{y}, y)$ from the start of the training complicates optimization since many scans have a large prediction deviation, which makes the optimization susceptible to vanishing gradients. $\mathcal{L}_{\text{MSE}}(\hat{y}, y)$, on the other hand, causes the optimization to prioritize large deviations, making this a suitable loss to use at the beginning of training. To combine the advantages of these two losses, a shifting loss function $\mathcal{L}_{\text{shift}}(\hat{y}, y)$ was used where the loss function changes over the epochs. This was done by starting with a $\mathcal{L}_{\text{combined}}(\hat{y}, y)$ with $\lambda_r = 0$ and $\lambda_m = 1$ for the first 100 epochs and linearly reducing λ_m to 0 between epochs 100 and 300, while linearly increasing λ_r from 0 to 1. The last 100 epochs were trained with $\lambda_r = 1$ and $\lambda_m = 0$.

The loss values over the epochs for different prediction errors are shown in Figure 8b. Note that this does not represent an actual training but rather the adjustment of loss weights during training. The difference in loss between large and small prediction errors is very large in the first 100 epochs when the $\mathcal{L}_{\text{MSE}}(\hat{y}, y)$ is used. This difference becomes smaller as λ_m is decreased and λ_r is increased until it is quite small for the last 100 epochs when only $\mathcal{L}_{\text{round}}(\hat{y}, y)$ is used.

3.3.4 Multi-class classification approach

The multi-class classification approach used the cross-entropy loss $\mathcal{L}_{\text{CE}}(\hat{y}, y)$, shown in Equation 4.

3.4 Training specifications

All networks were trained on an Nvidia A40 48GB GPU with 504 GB of RAM. An Adam optimizer with a learning rate of 10^{-4} was used. The training was done in batches of 12 scans per batch, where the model weights were updated after each batch. All models were trained for 400 epochs.

While training the model over 400 epochs, the weights of the last 50 epochs were stored and averaged. This was done to alleviate the effect of the fluctuations in the validation performance metrics during training.

3.5 Performance metrics

The resulting averaged models were evaluated in terms of accuracy and root-mean-squared error (RMSE). The accuracy is defined by the number of correctly predicted Carnegie stages divided by the total number of scans. Since the network output is continuous in the metric regression approach, the network output was first rounded to the nearest Carnegie stage. The RMSE is given by Equation 5, where N is the number of scans over which the metric is calculated.

$$\text{RMSE}(\hat{y}, y) = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (\hat{y}_n - y_n)^2} \qquad (5)$$

95% confidence intervals were calculated for both metrics by bootstrap resampling the validation or test set. This was done by randomly selecting samples from the validation or test set 10^3 times, with replacement, and calculating the metrics on these datasets. The 95% confidence intervals were computed using the percentile method described by Sanchez-Lengeling et al. [22], by finding the 2.5th and 97.5th percentile of the resulting metric values. The performance of two models was considered significantly different when their performance metrics fell outside each other's 95% confidence intervals.

4 Experiments

Experiment 1: loss function

To test whether a metric regression approach outperforms a multi-class classification approach, both approaches were implemented. The metric regression approach used the metric regression network output architecture and was trained with $\mathcal{L}_{\text{MSE}}(\hat{y}, y)$. The multi-class classification approach used the multi-class classification output architecture with 12 output channels and was trained with $\mathcal{L}_{\text{CE}}(\hat{y}, y)$, with $C = 12$.

The effect of using different loss functions was evaluated by comparing the performance of networks trained with different loss functions in the metric regression approach. These were $\mathcal{L}_{\text{MSE}}(\hat{y}, y)$, $\mathcal{L}_{\text{Huber}}(\hat{y}, y)$ with $\delta = 1$, $\mathcal{L}_{\text{round}}(\hat{y}, y)$, $\mathcal{L}_{\text{combined}}(\hat{y}, y)$, with $\lambda_r = 1$, $\lambda_m = 1$ and $\lambda_r = 1$, $\lambda_m = \frac{1}{10}$, and $\mathcal{L}_{\text{shift}}(\hat{y}, y)$, where $k = 5$ for all functions that used $\mathcal{L}_{\text{round}}(\hat{y}, y)$. The networks were evaluated on the validation set.

Experiment 2: embryonic size

To investigate the effect of incorporating the embryonic size in the model input, the isotropic voxel size of the input scans was given as secondary input to the network. Since the embryonic size is correlated to the voxel size of the preprocessed input image, this information is implicitly provided to the network. Different layer widths were used for the fully connected layers that take the DenseNet output and voxel size as input: $W_{FC1} = 1000$, $W_{FC2} = 500$; $W_{FC1} = 1000$, $W_{FC2} = 100$; and $W_{FC1} = 100$, $W_{FC2} = 50$.

Experiment 3: performance on test set

The best model from Experiment 1, based on the performance on the validation set, was evaluated on the test set. The same model was also trained on only stages 16-23 of the training set. This was done to assess the impact of stages 13-15, which have a limited amount of data available and lower scan resolution due to smaller embryos, on the model performance. The fetus class is also challenging since it contains more morphological variability, encompassing all cases beyond the Carnegie stages. Accordingly, the performance of this model was evaluated on a subset of the test set containing only stages 16-23 and compared to the performance on the same subset of the test set of the model trained on the entire training set.

Graphs were created by plotting the predicted Carnegie stages against the ground truth Carnegie stages for the test set. Additionally, to examine the relationship between prediction deviation and measured EV, these graphs were made for two subsets of the test set. These subsets were made by dividing the test set based on the EV for each ground truth Carnegie stage. Scans with an EV below the median per Carnegie stage were grouped into the "below-median EV" subset, while those with an EV above the median per Carnegie stage were assigned to the "above-median EV" subset.

Experiment 4: human performance

To provide context, we compared the model's performance to that of a human rater. A medical doctor, who was learning how to perform Carnegie staging in virtual reality, evaluated 46 embryonic scans ranging from stage 13 to 23 in a blind test. These ratings were then compared to the ratings of an expert rater and evaluated in terms of accuracy and RMSE.

Experiment 5: comparison to Carnegie stages literature

Although Carnegie stages are not based on length measurements, ranges of common values for the CRL are described by O'Rahilly and Müller [1]. The manually rated Carnegie stages of the ground truth labels were related to these CRL ranges by Rousian et al. [12], showing a generally good fit despite slightly higher CRL in 3D ultrasound-rated Carnegie stages. Likewise, we assessed whether the model's predicted Carnegie stages correlated with CRL similarly to the ground truth Carnegie stages. This was done by plotting the measured CRL against both ground truth and predicted Carnegie stages on the test set and comparing them to the CRL ranges per Carnegie stage described by O'Rahilly and Müller [1].

Experiment 6: linear regression using embryonic variables

To serve as a benchmark for comparing our ultrasound-based approach, a linear regression using embryonic variables was used to predict the Carnegie stage. Three distinct linear regression models were developed, fitting the Carnegie stage against GA, CRL, and $\sqrt[3]{\mathrm{EV}}$ within stages 13-23 of the training set. Cases labeled as fetuses were not used as the range of GA, CRL, and EV for these subjects was much wider. The regression models were evaluated on the test set.

Experiment 7: influence of scan quality

To evaluate the impact of ultrasound scan quality on model performance, a subset of scans in the test set underwent quality scoring. The same human raters responsible for Carnegie staging conducted this assessment. Scans were categorized as 'excellent', 'good', or 'moderate' depending on their effectiveness in facilitating Carnegie staging. The performance metrics of the best model were then compared across these three subsets.

Experiment 8: inspection of outliers

To gain insight into the causes of mispredictions by the model, scans in the validation and test sets for which the prediction deviation of the best model was greater than 1.5 were reconsidered by a human expert rater. These predictions were compared to the ground truth Carnegie stage and the model predictions. Additionally, the ultrasound scan quality and segmentation quality were evaluated to find possible causes of mispredictions.

5 Results

Experiment 1: loss function

The results for training with different loss functions are presented at the top of Table 1. When comparing the metric regression approach trained with $\mathcal{L}_{\mathrm{MSE}}(\hat{y}, y)$ to the multi-class classification approach trained with $\mathcal{L}_{\mathrm{CE}}(\hat{y}, y)$, there is no significant difference in accuracy. However, the metric regression approach demonstrates a significantly lower RMSE than the multi-class classification approach. This discrepancy is attributed to the multi-class classification's inability to differentiate between the severity of mistakes, leading to larger prediction errors and a higher RMSE.

Among all loss functions, $\mathcal{L}_{\mathrm{round}}(\hat{y}, y)$ exhibits the poorest performance. The network's inability to optimize effectively, likely due to a low gradient when the prediction error is substantial, results in it remaining at its initialization state.

We hypothesized that employing $\mathcal{L}_{\mathrm{Huber}}(\hat{y}, y)$ would lead the model to downplay the influence of outliers, potentially yielding higher accuracy at the expense of increased RMSE. However, this effect is not clearly observed. This observation applies similarly to $\mathcal{L}_{\mathrm{combined}}(\hat{y}, y)$ and $\mathcal{L}_{\mathrm{shift}}(\hat{y}, y)$.

The MSE loss, being the most straightforward option for metric regression, yielded the highest accuracy on the validation set. As none of the other losses surpassed its accuracy or RMSE, the MSE loss was selected as the optimal choice and used for subsequent experiments.

Table 1: Results metrics for different experiments. The accuracy and root mean squared error (RMSE) are given with their 95% confidence intervals. N represents the number of scans over which the metrics are calculated. When this is not specified, the metrics are calculated over the entire validation ($N = 296$) or test ($N = 272$) set.

Experiment	Result specification	Accuracy	RMSE
1: loss function	$\boldsymbol{\mathcal{L}_{\mathbf{MSE}}(\hat{y}, y)}$	**0.65 (0.59; 0.70)**	**0.65 (0.55; 0.78)**
	$\mathcal{L}_{\mathrm{CE}}(\hat{y}, y)$	0.61 (0.56; 0.66)	0.86 (0.74; 1.01)
	$\mathcal{L}_{\mathrm{Huber}}(\hat{y}, y)$	0.63 (0.56; 0.68)	0.64 (0.54; 0.76)
	$\mathcal{L}_{\mathrm{round}}(\hat{y}, y)$	0.00 (0.00; 0.00)	20.1 (19.8; 20.4)
	$\mathcal{L}_{\mathrm{combined}}(\hat{y}, y)$, $\lambda_m = 1$	0.61 (0.56; 0.67)	0.67 (0.57; 0.78)
	$\mathcal{L}_{\mathrm{combined}}(\hat{y}, y)$, $\lambda_m = 0.1$	0.58 (0.53; 0.64)	0.69 (0.59; 0.81)
	$\mathcal{L}_{\mathrm{shift}}(\hat{y}, y)$	0.61 (0.56; 0.67)	0.64 (0.55; 0.75)
2: embryonic size	$W_{FC1} = 1000$, $W_{FC2} = 500$	0.58 (0.53; 0.64)	0.66 (0.59; 0.75)
	$W_{FC1} = 1000$, $W_{FC2} = 100$	0.57 (0.52; 0.63)	0.68 (0.60; 0.78)
	$W_{FC1} = 100$, $W_{FC2} = 50$	0.63 (0.57; 0.68)	0.66 (0.57; 0.76)
3: performance on test set	**Entire set**	**0.59 (0.54; 0.65)**	**0.62 (0.55; 0.68)**
	Eval on stages 16-23 ($N = 242$)	0.60 (0.54; 0.66)	0.60 (0.53; 0.67)
	Train/eval on stages 16-23 ($N = 242$)	0.65 (0.59; 0.71)	0.56 (0.51; 0.61)
4: human performance	Intermediate rater ($N = 46$)	0.63 (0.50; 0.76)	0.65 (0.49; 0.80)
6: linear regression using embryonic variables	GA	0.39 (0.33; 0.45)	1.08 (0.99; 1.19)
	CRL	0.65 (0.60; 0.71)	0.93 (0.68; 1.17)
	$\sqrt[3]{\mathrm{EV}}$	0.61 (0.55; 0.67)	0.84 (0.65; 1.05)
7: influence of US scan quality	Total ($N = 138$)	0.57 (0.49; 0.65)	0.63 (0.56; 0.71)
	Excellent ($N = 45$)	0.60 (0.47; 0.73)	0.60 (0.49; 0.71)
	Good ($N = 63$)	0.56 (0.44; 0.68)	0.63 (0.52; 0.74)
	Moderate ($N = 30$)	0.57 (0.40; 0.73)	0.69 (0.52; 0.85)

Experiment 2: embryonic size

Table 1 presents the outcomes for the network architecture with voxel size as a secondary input, across various fully connected layer widths. None of the results surpass the performance of the $\mathcal{L}_{\mathrm{MSE}}(\hat{y}, y)$ network, which was trained without voxel size as a secondary input. This indicates that the addition of embryonic size as secondary input did not offer the model sufficient new information to improve its performance. Consequently, the $\mathcal{L}_{\mathrm{MSE}}(\hat{y}, y)$ model was selected as the optimal choice.

Experiment 3: performance on test set

The results on the complete test set of the model trained with $\mathcal{L}_{\mathrm{MSE}}(\hat{y}, y)$, as well as the results on stages 16-23 of the test set are displayed in Table 1. Additionally, the results of the model trained solely on stages 16-23 of the training set are provided.

The results on the test set and validation set are comparable, as the metrics fall within each other's 95% confidence intervals.

No significant performance increase is found when evaluating the model trained on the entire training set on only stages 16-23 of the test set. This can be explained by the small influence of the, often mispredicted, stages 13-15 on the evaluation metrics because of their small number of scans in the test set.

There is no significant improvement when comparing the evaluation on stages 16-23 of the model trained on the entire training set to that of the model trained solely on stages 16-23 of the training set.

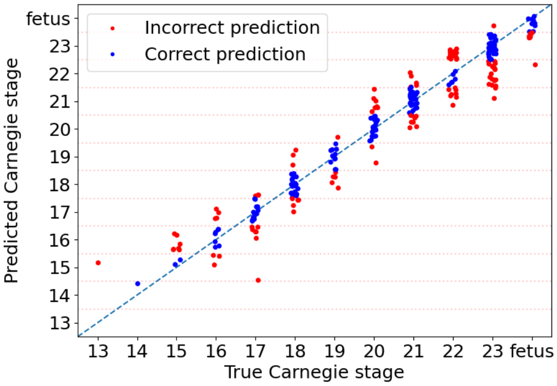


Figure 9: Graph depicting the model-predicted Carnegie stage plotted against the ground truth Carnegie stage for the test set.

The predicted Carnegie stages on the test set are plotted against the ground truth Carnegie stages in Figure 9. In this representation, it is apparent that the distribution of predictions is centered around the Carnegie stage label for the middle classes (stages 16-22). There is a bias towards higher classes for stages 13-15 and towards lower classes for stage 23 and the fetus class.

Figures 10a and 10b display plots of the predicted Carnegie stage against the ground truth Carnegie stage for the below-median EV and above-median EV per Carnegie stage subsets of the test set. In these plots, it is visible that prediction deviations are more biased toward lower stages for the small EV scans and toward higher stages for the large EV scans.

Experiment 4: human performance

The performance metrics for the human rater are shown in the corresponding row of Table 1. The found accuracy and RMSE, with 95% confidence intervals, are 0.63 (0.50; 0.76) and 0.65 (0.49; 0.80) respectively. These results are similar to the results of the CNN model, which had an accuracy of 0.59 (0.54; 0.65) and a RMSE of 0.62 (0.55; 0.68) on the test set.

Experiment 5: comparison to Carnegie stages literature

The measured CRL is plotted against the ground truth Carnegie stage (Figure 11a) and predicted Carnegie stage respectively (Figure 11b). The expected CRL ranges per Carnegie stage described by O'Rahilly and Müller [1] are included. The predicted Carnegie stages follow the same CRL distribution as the ground truth Carnegie stages. This is an indication that the model assigns Carnegie stages in a manner that relates to the CRL ranges in the same way as the ground truth.

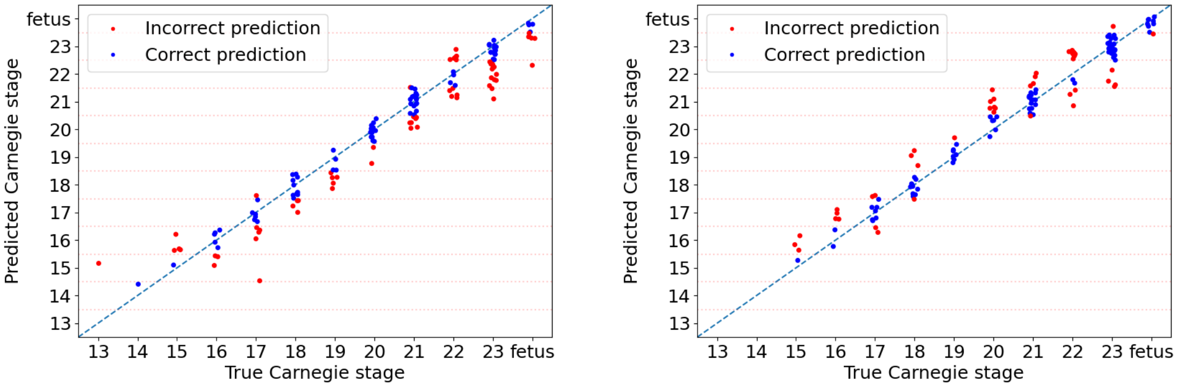
Experiment 6: linear regression using embryonic variables

The ground truth Carnegie stage is plotted against the GA, CRL, and $\sqrt[3]{\text{EV}}$ for stages 13-23 of the training set in Figure 12. Here, the linear fit is also plotted for each model.

The performance metrics for the regression models fitted on GA, CRL, and $\sqrt[3]{\text{EV}}$ are displayed in Table 1. The fitted trained on CRL and $\sqrt[3]{\text{EV}}$ outperform the model fitted on GA in accuracy. The model fitted on CRL outperforms the CNN model in accuracy, however, it performs worse than the CNN model in RMSE. The CNN model outperforms the model fitted on $\sqrt[3]{\text{EV}}$ in RMSE.

Experiment 7: influence of US scan quality

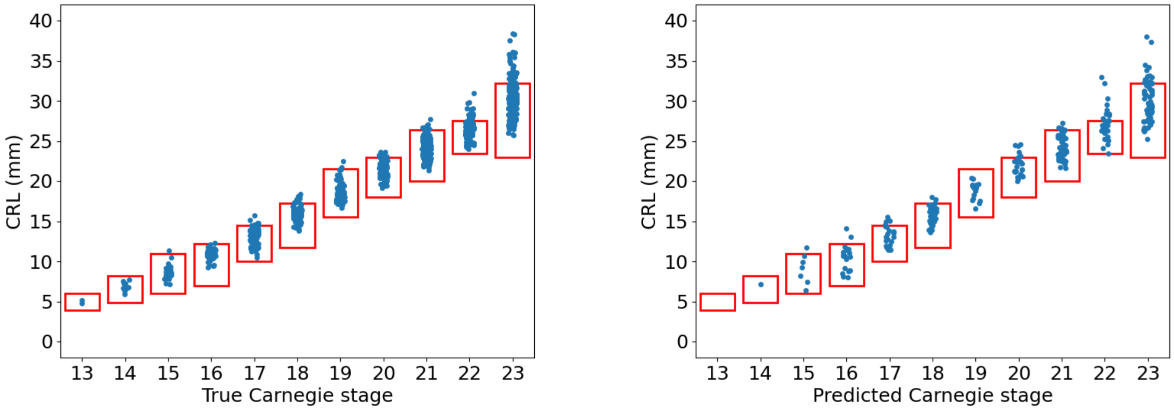
Table 1 presents the performance metrics for three subsets of the test set categorized by ultrasound scan quality: excellent, good, and moderate. Additionally, the performance metrics for all scored scans are included. Based on the confidence intervals, we cannot conclude that the model's performance differs across subsets. However, there appears to be a noticeable trend, particularly regarding the RMSE, which decreases as scan quality improves. This suggests a correlation between model performance and scan quality.



(a) Below median EV subset

(b) Above median EV subset

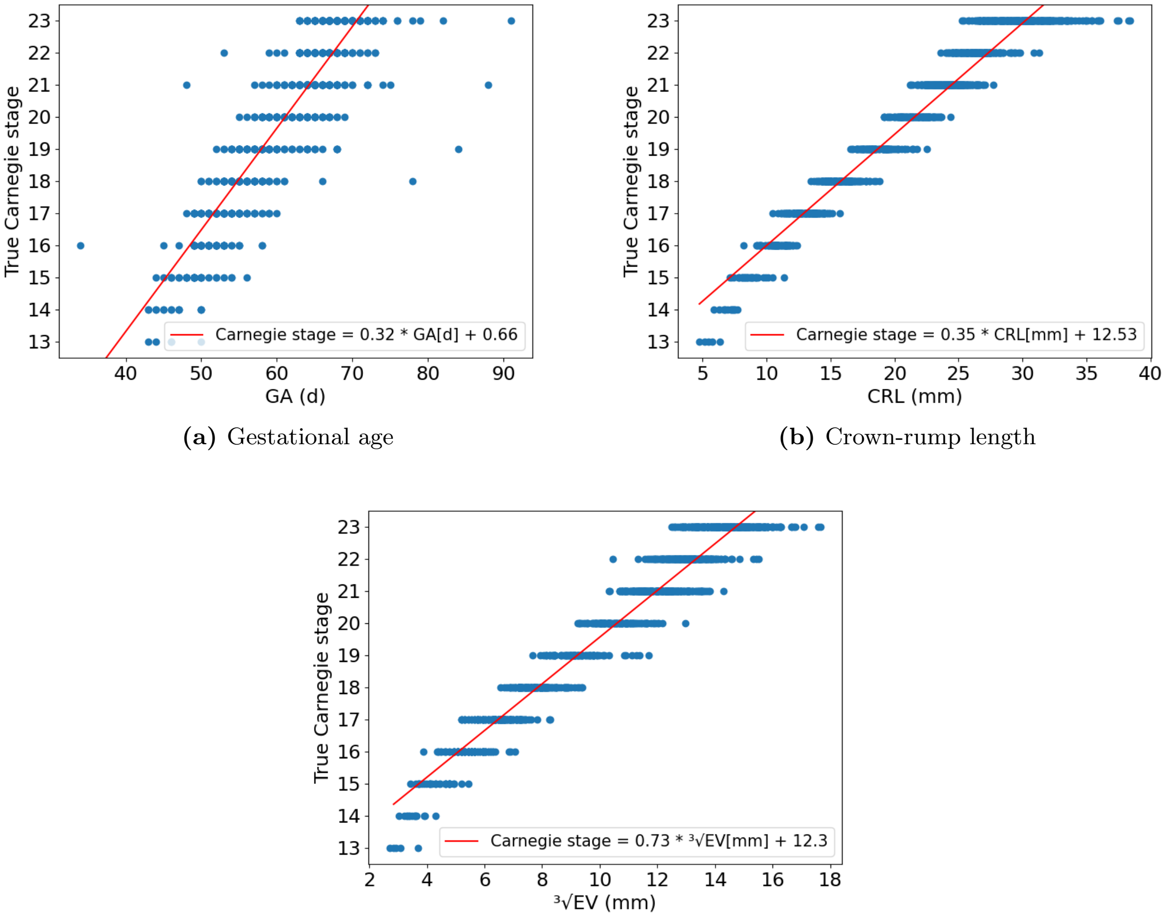
Figure 10: Graphs depicting the predicted Carnegie stage by the CNN model plotted against the ground truth Carnegie stage for two subsets of the test set. The below median EV set comprises scans with the lowest half of EV per Carnegie stage, while the above median EV set includes scans with the highest half of EV per Carnegie stage.



(a) Ground truth Carnegie stage

(b) Predicted Carnegie stage

Figure 11: Plot of measured CRL against ground truth Carnegie stage and predicted Carnegie stage. The expected CRL ranges per Carnegie stage described by O'Rahilly and Müller [1] are indicated with the red boxes.



(a) Gestational age

(b) Crown-rump length

(c) Embryonic volume

Figure 12: Ground truth Carnegie stage plotted against embryonic measures for stages 13-23 of the training set. The linear regression model fitted on the measures is included as the red line. The formula that describes the linear fit is indicated in the legends.

Experiment 8: inspection of outliers

Table 2 displays the outcomes of the second rating of outliers in both the validation and test sets. The second rating corresponded with the first rating in four instances, corresponded with the CNN model once, and fell between ground truth and CNN model prediction five times. Among the outliers, nine scans had poor scan quality. These were characterized by a low contrast-to-noise ratio (CNR) six times and a low resolution five times. Issues with segmentation were observed, including six instances of missing parts of the limbs, two cases of including part of the yolk sac, and one instance of missing the embryo completely.

6 Discussion

The Carnegie staging system facilitates the assessment of embryonic development in terms of morphology during a critical period in its development. However, manual in-utero Carnegie staging using 3D ultrasound scans is time-consuming. To overcome this, we proposed an automated Carnegie staging method using deep learning. A 3D MONAI DenseNet-121 was trained on a dataset of 1413 embryonic 3D ultrasound scans labeled with Carnegie stages obtained from the Rotterdam Periconceptional Cohort. Embryonic volumes were automatically segmented from the data. Segmented scans were cropped and resized to a standard dimension to prevent the model from using the embryonic size in its prediction.

The model trained with the MSE loss emerged as the best model. This model gave an accuracy of 0.59 (0.54; 0.65) and a RMSE of 0.62 (0.55; 0.68), with 95% confidence intervals, on the test set. This performance was similar to the performance of an intermediate human rater, who achieved an accuracy of 0.63 (0.50; 0.76) and a RMSE of 0.65 (0.49; 0.80) on a selection of 46 scans.

To assess possible causes of mispredictions by the CNN model, the model was tested on subsets of ultrasound scans with image quality visually rated from excellent to moderate. A trend was observed that suggests that increased scan quality decreased the model RMSE, indicating that poor scan quality was a possible cause of mispredictions. This is supported by the fact that, upon evaluating the outlier scans, nine out of ten were classified as having poor scan quality.

Six out of ten outlier scans had missing parts of the limbs in the embryonic segmentation. Given the significance of limbs in Carnegie staging using 3D ultrasound [12], accurate limb segmentation likely plays a pivotal role. Therefore, it would be valuable to further investigate the correlation between segmentation quality and prediction performance, as well as to establish a measure for segmentation quality.

Furthermore, it is noteworthy that only 20% of scans labeled as stage 22 are correctly classified, whereas 53% are categorized as stage 23. Additionally, 28% of scans labeled as stage 23 are misclassified as stage 22. This considerable overlap between the predictions of these two stages aligns with Carnegie staging by human observers, who often find distinguishing between these two stages particularly challenging. This is partly because the common GA range for these stages overlap: 54-58 days for stage 22 and 56-60 days for stage 23, see Figure 1 [6]. Furthermore, the common CRL range for stage 22 lies completely within the common CRL range for stage 23, as can be seen in Figure 11.

Although the model performance was compared to the performance of a human rater who was learning how to perform Carnegie staging in virtual reality, an exact measure for the inter-observer variability of the human raters who established the ground truth dataset is lacking. The fact that the second rating of the outlier scans coincided with the first rating only four out of ten times, emphasizes the importance of knowing the inter-observer variability. Addressing this limitation could involve having the raters determine the Carnegie stage for multiple scans in a double-blind experiment. This knowledge could provide insights into the maximum achievable accuracy for the model.

Various training strategies incorporating different loss functions and network output architectures were explored. Custom loss functions, designed to increase the gradient at a prediction error of 0.5 and potentially enhance accuracy at the cost of an increased RMSE, were employed. However, none of

Table 2: Results of the second rating by a human expert rater of outliers in the validation and test set. The scan and segmentation quality evaluation are included, where the scan quality is characterized by the contrast-to-noise ratio (CNR) and resolution (res).

Carnegie stage			**Scan quality**	**Segmentation quality**
Ground truth	**CNN**	**Second rating**		
13	15.19	14	poor (low CNR, res)	part yolk sac segmented
13	15.14	14	poor (low CNR, res)	part yolk sac segmented
14	15.66	15	poor (low res)	part of limbs missing
14	15.65	14	moderate	good
17	14.53	17	poor (low res)	part of limbs missing
18	13.53	17	poor (low CNR)	embryo not in segmentation
21	17.91	21	poor (low res)	part of limbs missing
23	21.47	22	poor (low CNR)	part of limbs missing
23	21.10	23	poor (low CNR)	part of limbs missing
fetus	22.31	22	poor (low CNR)	part of limbs missing

these custom loss functions demonstrated improved performance compared to the default MSE loss. One potential explanation for this outcome is that while the custom loss functions were designed to influence network optimization, their impact may have been overshadowed by broader challenges related to generalization to unseen data. Factors such as dataset variability and the complexity of the task may have had a much larger impact on the network optimization, limiting the impact of the custom loss functions in improving the accuracy.

When assessing the best model by selecting per Carnegie stage the below- and above-median EV subsets of the test set, a trend was observed. Predictions tend to be biased towards lower stages for the below-median EV subset and towards higher stages for the above-median EV subset. This phenomenon suggests that embryonic size plays a role in influencing the model's predictions. Although the preprocessing steps aimed to blind the model to information about embryonic size, it cannot be definitively stated that this information is entirely excluded from the input. Given that cropped scans of smaller embryos generally have fewer voxels, resulting in increased oversampling compared to larger embryos, the information is distributed across a greater number of voxels. This oversampling effect could have been leveraged by the model to infer embryonic size.

Another possible explanation for the observed correlation between EV and prediction deviation lies in the creation of the ground-truth labels. In case the human rater misclassified a scan in the test set to be one stage higher/lower, the ground truth of this scan is one stage too high/low, resulting in the EV for that stage being lower/higher than the median EV for that stage. If the CNN model correctly classifies this scan, it is below/above the inaccurate ground truth. As such, a scan that is above the ground truth is likely to be in the above median subset and vice-versa. Therefore, we cannot definitively conclude whether embryonic size influenced the model predictions, however, it is likely that this information was used to some degree.

Incorporating embryonic size into the model input did not lead to improved performance. This could mean that the model was already able to find the embryonic size from the input scans and exploited its correlation to Carnegie stages for its predictions. Another explanation for the lack of performance improvement is that the correlation between the Carnegie stage and embryonic size was insufficient to further improve the model performance. This idea is supported by noticing that linear regression models trained on CRL and EV did not surpass the performance of the model trained solely on ultrasound scans.

Particularly for stages 22 and 23, the measured CRL was large compared to the ranges described by O'Rahilly and Müller [1] for both the ground truth and the model predictions. This indicates a potential discrepancy between the ground truth and Carnegie stages literature. This could be caused by the absence of histological examinations of the embryos, which form part of the basis for Carnegie staging, as highlighted by Rousian et al. [12].

While Carnegie stage predictions were centered around true stages, mispredictions were more common towards higher stages for earlier stages and towards lower stages for later stages. This trend likely stems from the regression approach implemented, where predictions towards the middle classes generally result in lower losses compared to predictions towards the outer classes. This causes the model to be biased towards the middle classes since this generally has a lower risk of misprediction.

Although the Carnegie stages are categorical, representing distinct developmental stages, our model's output gives a continuous value that represents the Carnegie stage. Embryonic development is a continuous process, where the Carnegie stages represent selected points on this evolving timeline [5]. Using the continuous output of the model may provide a more nuanced representation of the changing embryonic morphology, requiring further investigation.

Notably, only a few scans displayed prediction deviations larger than one stage in the test set, leading to an accuracy of 0.98 when a one-stage deviation is allowed. This suggests that the model could be useful for identifying statistical correlations in large datasets, using for example the aforementioned continuous alternative for Carnegie stages, where precise Carnegie staging is not essential.

The inconsistency of embryo orientation in the input scans stemmed from the natural variation of embryo orientation in the uterus. This challenge was addressed with data augmentation by random multiples of 90-degree rotations in all three dimensions to enhance the model's robustness to embryo orientation. However, this variation could be eliminated through embryonic alignment techniques, potentially enhancing model performance. This is because correct alignment causes better overlap in the dataset of structures such as the brain ventricles and limbs, making it easier to identify these structures.

The under-represented classes in the dataset were oversampled to ensure a balanced presence of samples from all classes in the training set, mitigating the model's bias towards over-represented classes. However, this adjustment likely led to a decrease in model performance on the over-represented classes. This trend is visible in the results of the model trained solely on stages 16-23, which, while not surpassing the model trained on the entire dataset according to our significance criteria, exhibits a trend indicating this effect. Exploring alternative methods like cost-sensitive learning, where costs are assigned to misclassifications of different classes, may help alleviate this issue [23].

Additionally, a limitation exists concerning the data augmentation process, which allows for 64 possible augmentation operations. In each epoch, the smallest classes are oversampled to a greater extent, resulting in the presence of duplicates in the dataset. This challenge could be mitigated by incorporating a more diverse range of augmentation techniques and ensuring that each scan receives a unique augmentation.

Another limitation involves the limited interpretability of the model, as it is unclear which exact information is used for the model predictions. To improve interpretability, one potential approach is to have the model base its predictions on distinct parts of the embryo. This could be achieved by segmenting particular areas of the embryo, such as the limbs or head, and training the model specifically on these regions. Such an approach could give insights into the Carnegie stage assigned by the model to individual embryo parts, which could be compared with the features used by a human rater.

Another potential solution to enhance interpretability is to employ a deep ordinal regression framework described by Niu et al. [10]. In this framework, the network predicts, for each Carnegie stage, whether an input scan belongs to a higher stage than that Carnegie stage or not. This modification makes the network output more informative by incorporating a measure for uncertainty.

Alternatively, employing a machine learning approach, such as support vector regression, with carefully selected image-based features could also improve interpretability. These features should match those used in Carnegie staging by human observers, for instance, the curvature of the trunk or the shape of the limbs and brain ventricles. By doing so, it becomes clearer which information the model uses for its prediction, thus enhancing interpretability. Moreover, this strategy may help mitigate potential biases associated with embryonic volume.

7 Conclusion

This study marks a significant step towards automating Carnegie staging in 3D ultrasound. The 3D MONAI DenseNet-121 trained with the Mean Squared Error (MSE) loss emerged as the most effective. The model achieved an accuracy of 0.59 and a Root Mean Squared Error (RMSE) of 0.62 on the test set. This was comparable to an intermediate human rater, who achieved an accuracy of 0.63 and a RMSE of 0.65. Challenges related to segmentation quality, inter-observer variability, and model interpretability require further investigation to improve automated Carnegie staging. Furthermore, exploring the application of this model in research or clinical practice could enhance the assessment of normal and abnormal development during the critical embryonic period.

Acknowledgments

We thank all participants of the Rotterdam Periconceptional cohort (Predict study), without whom this study could not have been performed. Further, we would like to thank the Rotterdam Periconceptional Cohort team for their support in the processes of recruitment and data collection, which contributed to the results of this study.

References

1. O'Rahilly, R. & Müller, F. Developmental Stages in Human Embryos: Revised and New Measurements. *Cells Tissues Organs* **192,** 73–84. ISSN: 1422-6405 (2010).
2. Steegers-Theunissen, R. P. M. *et al.* Cohort profile: the Rotterdam periconceptional cohort (predict study). *International Journal of Epidemiology* **45,** 374–381 (2016).
3. Verwoerd-Dikkeboom, C. M., Koning, A. H. J., van der Spek, P. J., Exalto, N. & Steegers, E. A. P. Embryonic staging using a 3D virtual reality system. *Human Reproduction* **23,** 1479–1484. ISSN: 0268-1161 (2008).
4. Parisi, F. *et al.* Periconceptional maternal one-carbon biomarkers are associated with embryonic development according to the Carnegie stages. *Human Reproduction* **32,** 523–530. ISSN: 0268-1161 (2017).
5. Hill, M. A. Early human development. *Clinical obstetrics and gynecology* **50,** 2–9 (2007).
6. Flierman, S., Tijsterman, M., Rousian, M. & de Bakker, B. S. Discrepancies in Embryonic Staging: Towards a Gold Standard. *Life* **13,** 1084. ISSN: 2075-1729 (2023).
7. Rousian, M. *et al.* An innovative virtual reality technique for automated human embryonic volume measurements. *Human Reproduction* **25,** 2210–2216. ISSN: 0268-1161 (2010).
8. Anwar, S. M. *et al.* Medical Image Analysis using Convolutional Neural Networks: A Review. *Journal of Medical Systems* **42,** 1–13. ISSN: 1573-689X (2018).
9. Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. Densely Connected Convolutional Networks. *ArXiv e-prints.* eprint: 1608.06993 (2016).
10. Niu, Z., Zhou, M., Wang, L., Gao, X. & Hua, G. Ordinal Regression with Multiple Output CNN for Age Estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 27–30. ISSN: 1063-6919 (2016).

11. Rousian, M. *et al.* Cohort profile update: the Rotterdam Periconceptional Cohort and embryonic and fetal measurements using 3D ultrasound and virtual reality techniques. *International Journal of Epidemiology* **50,** 1426–1427l (2021).

12. Rousian, M. *et al.* First trimester brain ventricle fluid and embryonic volumes measured by three-dimensional ultrasound with the use of I-Space virtual reality. *Human Reproduction* **28,** 1181–1189 (2013).

13. Parisi, F. *et al.* Effect of human embryonic morphological development on fetal growth parameters: the Rotterdam Periconceptional Cohort (Predict Study). *Reproductive BioMedicine Online* **38,** 613–620. ISSN: 1472-6483 (2019).

14. Verwoerd-Dikkeboom, C. M. *et al.* Innovative virtual reality measurements for embryonic growth and development. *Human Reproduction* **25,** 1404–1410. ISSN: 0268-1161 (2010).

15. O'rahilly, R., Muller, F. & Streeter, G. L. Developmental stages in human embryos: including a revision of Streeter's Horizons and a survey of the Carnegie Collection (1987).

16. Hoek, J. *et al.* Periconceptional maternal and paternal homocysteine levels and early utero-placental (vascular) growth trajectories: The Rotterdam periconception cohort. *Placenta* **115,** 45–52. ISSN: 0143-4004 (2021).

17. Bastiaansen, W. A. P. *et al.* EP13.03: Automatic volumetric measurements of the embryo and head during the first trimester using artificial intelligence: the Rotterdam Periconception Cohort. *Ultrasound in Obstetrics & Gynecology* **62,** 166. ISSN: 0960-7692 (2023).

18. Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J. & Maier-Hein, K. H. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods* **18,** 203–211. ISSN: 1548-7105 (2021).

19. Moradi, R., Berangi, R. & Minaei, B. A survey of regularization strategies for deep models. *Artificial Intelligence Review* **53,** 3947–3986. ISSN: 1573-7462 (2020).

20. Ruiz, P. *Understanding and visualizing DenseNets* ISBN: 978-768809239 (Towards Data Science, 2018).

21. Wang, J. *et al.* Gray Matter Age Prediction as a Biomarker for Risk of Dementia. *Proceedings of the National Academy of Sciences* **116,** 21213–21218 (2019).

22. Sanchez-Lengeling, B. *et al.* Machine learning for scent: Learning generalizable perceptual representations of small molecules. *arXiv preprint arXiv:1910.10685* (2019).

23. Buda, M., Maki, A. & Mazurowski, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* **106,** 249–259. ISSN: 0893-6080 (2018).

Appendix

Automated Carnegie Staging of the Human Embryo in 3D Ultrasound using Deep Learning: A Literature Study

Ruben Niemantsverdriet

December 2023

Abstract

The majority of reproductive failures, pregnancy complications, and adverse pregnancy outcomes stem from the periconceptional period. This is the period that spans from 14 weeks before conception to 10 weeks after conception. This period contains the embryonic period in which most of the organs in the human body develop rapidly. The Carnegie staging system consists of 23 stages that describe the entire embryonic period based on the morphological characteristics of the embryo. The Carnegie stage is used as an alternative measure for embryonic development during this critical period, next to embryonic size and age. A convolutional neural network will be trained on a dataset of labeled 3D ultrasound scans of embryos to perform in-utero Carnegie staging. This report gives an overview of relevant literature about Carnegie staging in 3D ultrasound. It was found that the most important embryonic features for staging are limb development, trunk curvature and brain ventricle development. Furthermore, this report gives an overview of often-used regularization methods to improve training a neural network with limited data. Data augmentation and batch normalization seem like the most promising regularization techniques for this project. Finally, the study addresses datasets with ordered labels. It is suggested that this is done most efficiently by implementing a deep ordinal regression framework, using either conditional training sets or soft ordinal vectors as labels.

1 Introduction

The pregnancy is divided into an embryonic and a fetal period. During the embryonic period, which consists of the first 8 weeks of pregnancy, many features, including most of the organs in the human body, start to develop rapidly [1]. The Carnegie staging system facilitates the assessment of normal embryonic development in terms of morphology. Additionally, it enables the identification of abnormalities [2]. The Carnegie staging system describes the entire embryonic period based on the internal and external physical characteristics of the embryo, e.g. limbs and brain ventricle development. It comprises 23 stages starting at a fertilized oocyte (stage 1) and ending with an embryo in which all essential internal organ systems are present (stage 23), marking the end of the embryonic period and the start of the fetal period. The system is based solely on the morphological characteristics of the embryo and not on embryonic size or age [3].

The embryonic period, that the Carnegie stages describe, is part of the periconceptional period, which spans from 14 weeks before conception to 10 weeks after conception. The majority of reproductive failures, pregnancy complications, and adverse pregnancy outcomes stem from this period [4]. The Rotterdam Periconceptional Cohort was initiated as a cohort study to investigate the effects of maternal and paternal health during this critical period of pregnancy [5]. In this study, transvaginal 3D ultrasound scans of the embryo are made between 6 and 12 weeks of gestational age [5]. Currently, in-utero Carnegie staging is done by human raters using a virtual reality 3D projection system, which is time-consuming [2, 6]. Therefore, an automated Carnegie staging method would be helpful for more efficient staging.

To address this, a deep learning method was studied that predicts the Carnegie stage directly from the 3D ultrasound scan. This was based on a deep convolutional neural network (CNN) since this has shown good performance in a wide range of applications [7]. This network was trained on a dataset that consists of 1413 labeled 3D ultrasound scans, encompassing stage 13 to beyond stage 23, in which subjects are labeled as fetuses. This method should preferably give some insights into the embryonic features that are important for the staging. On top of that, the model should base its predictions only on the embryo's morphology, not size or age. In this way, the Carnegie stage can be compared to other embryonic variables, like size or age, as an independent characteristic of embryonic development.

For this project, only limited labeled data is available. A small dataset can reduce the performance of a neural network as the network has many parameters to optimize. To mitigate this issue, regularization techniques are employed. Regularization techniques introduce a bias into the model by incorporating knowledge about parameters that the model cannot effectively learn due to the limited available data [8]. In general, any component of the model that alleviates the problem of limited data is called regularization. Hence, this literature study provides an overview of relevant regularization techniques.

Finally, we address the question of whether to treat the problem as a classification or a regression problem. This problem arises from the fact that Carnegie stages are ordered and not independent classes, as is assumed in a multi-class classification approach. However, the scale can not be considered continuous, as is assumed in a typical regression approach. An overview is given of literature that deals with this issue of datasets with ordered classes.

2 Methods

The first objective of this literature study is to provide an overview of relevant literature about the Carnegie staging system and how the staging is done using 3D ultrasound. This is done with the goal of better understanding how the ground truth Carnegie stage labels are determined. First, some information about discrepancies in the use of Carnegie stages is described. Then, literature about Carnegie staging in 3D ultrasound is explored. Lastly, relevant embryonic features for Carnegie staging in 3D ultrasound are discussed.

Secondly, a literature search was done for deep learning regularization techniques to alleviate the issue of data shortage. This was done to create a broad overview of regularization techniques for similar problems as in this project. Papers that described an overview of techniques were searched in Google Scholar with the search words: "review deep learning regularization". Only papers published since 2017 were considered. Techniques that were described in at least two of these review papers were investigated further and are described in more detail.

Finally, this literature study addresses the challenge of effectively utilizing datasets with ordered classes in deep learning classification/regression tasks. First, a brief overview of the more standard multi-class classification and metric regression strategies is given. After this, an overview of techniques for performing deep ordinal regression is given. For this, literature was found in Google Scholar with search words: "ordinal regression AND deep learning". Papers were selected based on number of citations (>100) or relevance.

3 Carnegie staging system

3.1 Discrepancies in Carnegie staging

After multiple iterations since the beginning of the 20th century, the currently used Carnegie staging system was described by Ronan O'Rahilly and Fabiola Müller in 1987 [9]. As such, it became the first widely recognized human embryonic staging system [10]. Although the Carnegie staging system is used universally for the staging of ex vivo human embryos, literature concerning the distinctions between Carnegie stages is inconsistent [10]. While embryonic age and length are not the main features on which the staging system is based, one would expect to find a consistent mean age and length for each stage in the literature; however, this is not the case.

Possible reasons for this could be the differences in the interpretation of normal development between observers. Furthermore, uncertainty exists in determining the embryonic age, since the exact moment of fertilization is often difficult to pinpoint.

Although room for improvement still exists, Flierman et al. [10] propose Hill's 2007 paper [3] as the gold standard for embryological staging because of the use of embryonic samples from multiple collections and modern technologies and data collection techniques.

3.2 Carnegie staging in 3D ultrasound

Verwoerd-Dikkeboom et al. [2] have shown to accurately stage embryos using 3D ultrasound scans displayed in a 3D virtual reality system. This was done by only examining external morphological features, mainly limb development. O'Rahilly and Müller [1] raise the concern that using only external features has serious limitations. This is because it incorrectly uses a system designed for both internal and external features and is not precise enough for accurate staging. O'Rahilly and Müller [1] therefore claim that, in most cases, only an approximation of the Carnegie stage can be determined without considering internal features. Rousian et al. [6] staged embryos based on both internal and external features in a 3D virtual reality system. The limb development and torso curvature were used as external features, while the brain ventricle development was primarily used as internal features.

3.3 Features describing Carnegie stages

Examples of embryos of all the 23 stages are shown in figure 1. Since the Carnegie stages in our dataset are based on 3D ultrasound data, we only focus on 3D ultrasound features for stages 13-23. These features are based on the review paper by Flierman et al. [10] and are described in more detail below.

One of the features of Carnegie staging in 3D ultrasound is limb development [6]. The upper limbs start appearing at stage 12, where they are only small bulges. The same holds for the lower limbs, although they are usually slightly behind in development compared to the upper limbs. The upper and lower limbs become better visible at stage 14, as is indicated in figure 1. Throughout the following stages, both limbs grow further and develop more features, such as a handplate/footplate, fingers/toes, and an elbow/knee. Simultaneously, the length of the limbs and their position and orientation continue to develop in the subsequent stages.

Another important feature is the curvature of the trunk [6]. As can be seen in figure 1, the trunk is very curved at stage 13 and becomes less curved during the next stages.

Lastly, the brain ventricle development is used as the main internal feature [6]. Here, the distinction between stages mainly lies in the size comparison between the different brain ventricles.

4 Regularization techniques for deep learning

CNNs have proven to give promising results for a wide variety of medical image classification tasks [7]. Many CNN-based methods for medical image classification had to deal with data shortage since data annotation is often time-consuming and costly [11]. Techniques that are used to enhance the performance of a model on an unseen test set in the presence of data shortage are called regularization techniques [8]. Regularization aims to improve the generalization to unseen data and minimize the risk of overfitting on the training data. A wide variety of regularization techniques exist. The surveys by Moradi et al. [8], Kukacka et al. [12] and Nusrat and Jang [13] were used to get an overview of the most popular regularization methods. Techniques that are described

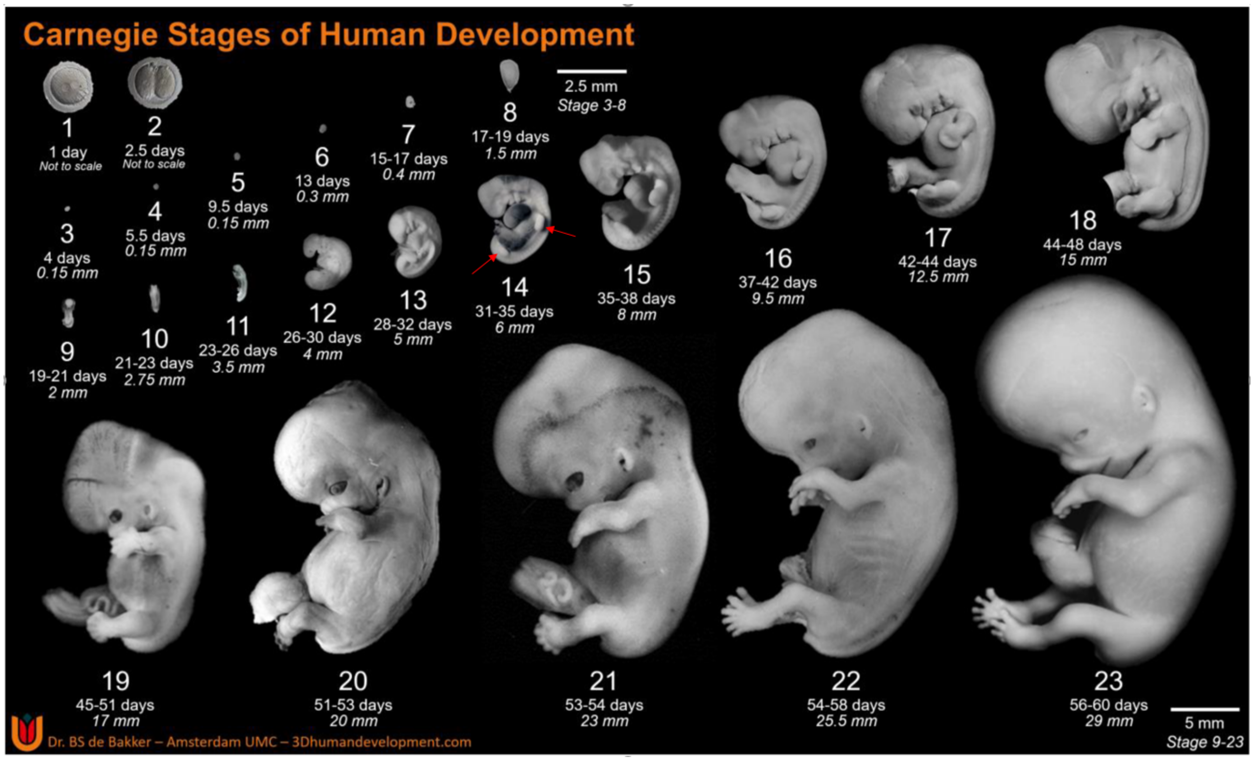


Figure 1: Illustrative images showcasing embryos at all 23 Carnegie stages obtained from Flierman et al. [10]. Microscope images were used for stages 1 and 2, 3D reconstructions based on histological sections for stages 3-8, and photos from subjects in the Carnegie collection for stages 9-23. Additionally, the mean embryonic age and length for each stage are provided. Red arrows indicate the upper and lower limbs in stage 14.

in at least two of these three review papers are discussed separately below. These are data augmentation, weight decay, regularization by network architecture (e.g. weight sharing, dropout, noise injection and multi-task learning), batch normalization, weight initialization and early stopping.

4.1 Data augmentation

Data augmentation involves modifying images in the training set to generate additional representative samples. It is a very popular way of improving the performance of a deep learning model [12, 14]. The modifications are carried out in a manner that mimics differences in the acquisition and anatomical variances among patients [8, 14]. This is done to prevent the model from concentrating on overly precise characteristics from the initial training set, thereby making it more resilient to variations in the training data. Simultaneously, it helps to enhance the model's overall ability to generalize.

Data augmentation can also help with class imbalance, where one or more classes are under-represented in the dataset, potentially causing the model to have a bias towards the over-represented classes. By using data augmentation to create more training samples for the under-represented classes, this effect can be reduced.

Three main categories for data augmentation techniques exist: basic, deformable and deep learning augmentation techniques. These categories are discussed below.

Basic augmentation techniques: Several basic techniques can be used to modify images and create new training samples [14, 15]. These techniques include geometric transformations like zooming, translation, flipping, and rotation. Another technique is cropping, where random patches of the image are selected to create new inputs. Occlusion is another technique, where random patches are removed from an image. Additionally, pixel intensity operations such as modifying the brightness or contrast can be used. Noise injection and filtering or mixing original images to create new ones are also common techniques. The latter can be done by simply adding two images together.

Deformable augmentation techniques: Deformable augmentation techniques are different from basic augmentation techniques as they do not maintain the shape of objects [14]. In effect, this creates even more variability in the data augmentation process. There are several types of deformable augmentation techniques, including randomized displacement field, spline interpolation, deformable image registration, and statistical shape models. A randomized displacement field is a technique that randomly shifts each pixel in an image. Spline interpolation, on the other hand, uses a piecewise polynomial function to interpolate new values between existing data points. Deformable image registration is a method that can be used to map existing images to create new ones. Statistical shape models utilize the shape variability of objects in the dataset to create deformations of existing images.

Deep learning-based augmentation techniques: Deep learning-based augmentation techniques learn the representation of the original data to create new synthetic data [14]. These techniques usually use a generative adversarial network (GAN), which consists of a generator and a discriminator [16]. The generator is a deep network that tries to map a fixed random, generally unstructured, distribution to the distribution of the target data. The discriminator, on the other hand, aims to estimate whether the samples that are presented are truly drawn from the training distribution or artificially generated by the generator, so if they are ‘real’ or ‘fake’. The discriminator is trained in such a way that it is encouraged to correctly classify real and fake images, while the generator is encouraged to generate samples that the discriminator incorrectly classifies as real. This competition causes both networks to improve each other’s performance. Eventually, the generator should be able to generate realistic synthetic data for the target data distribution, while the discriminator is unable to discriminate real from synthetic data. By sampling from a random distribution as input to the generator, it is possible to generate realistic synthetic data.

Perez et al. [17] found that, although deep learning-based data augmentation techniques are promising, they do not perform much better than traditional augmentation techniques, while consuming much more computing time. Furthermore, it is important to note that the type of data augmentation used should match the existing variation present in the dataset, as noted by Kukacka et al. [12]. One should thus inspect the available data and use data augmentation appropriately. Therefore, for this project, it is necessary to examine the input data and identify the variations present in it, such as ultrasound quality and acquisition, anatomical variations, and orientation and position of the embryo in the input scan and choose a data augmentation technique accordingly.

4.2 Weight decay

Another commonly used regularization technique is weight decay. This technique uses a penalty function applied to the model weights to constrain the model’s capacity [8]. The most used versions are the L_1 penalty norm, which takes the absolute magnitudes of the weights, and the L_2 penalty norm, which takes the squared magnitudes. Generally, L_1 regularization is used to guide the model to more sparse solutions, reducing the computational costs, while L_2 regularization tends to work better for reducing the risk of overfitting on the training data. Nusrat and Jang [13] found that L_1 regularization, although commonly used as a regularization technique, is not as effective as data augmentation and batch normalization.

4.3 Network architecture

Apart from data manipulation and penalty terms, regularization techniques can also be implemented in the network architecture. Different strategies for this are discussed below.

Choosing the right model complexity: One way to perform regularization in the network architecture is by reducing the model complexity. Model complexity denotes the capacity of models to effectively approximate complex distribution functions and the complexity of the functions that are represented by the model [18]. Less complex models generally have fewer parameters to learn, for instance, the number of filters in a CNN [19]. It is generally thought that less complex models tend to be less prone to overfitting with small datasets. Brigato et al. [19] found that less complex models indeed generalize better when trained with smaller datasets and tend to overfit less on the training data. However, they found that more complex models start performing better when using basic data augmentation.

Weight sharing: Another way of making a model less complex is by reusing several trained parameters throughout the network, known as weight sharing [12]. By imposing the network to use the same weights in multiple locations in the network, the amount of trainable parameters decreases, causing the network to be more robust and reducing the risk of overfitting. Weight sharing is standardly used in CNNs in the form of convolution kernels, which are applied to every part of the input image with the same weights [8].

Dropout: Dropout comprises the deactivation of randomly chosen neurons during training to prevent the decision-making process of the model from being dominated by a single feature [20]. By doing so, dropout performs a model ensembling by averaging over the smaller subnetworks that are created by the randomly (de)activated neurons [21, 22]. This prevents the neurons in the network from becoming too dependent on each other, which can lead to overfitting. Srivastava et al. [21] found that using dropout in training significantly increases generalization to unseen data in a wide variety of classification problems. Furthermore, they showed that dropout works better than other regularization methods, such as the L_1 and L_2 penalty norm.

Noise injection: Another way of inducing variability in the model, next to dropout, is by injecting random noise into various parts of the model during training [12]. This can be done by injecting noise in the input data, as mentioned in section 4.1, or by adding noise to the model weights or activation layers [23]. Similar to data augmentation, this helps the network to be more robust against input variations and thus functions as a regularization technique. He et al. [23] show that using a deep neural network with Gaussian noise injection at each layer yields improved accuracy on both clean and perturbed input test data. Moradi et al. [8], on the other hand, did not see any improvement in generalization by adding noise to either the input or the weights of the model.

Multi-task learning: Multi-task learning is a method in which the network aims to solve multiple tasks at once. The idea behind this concept is that the model generalization improves by having the model solve both the original task and related tasks [24]. This is because the representations learned for the related tasks are often useful for the primary task. This way, multi-task learning helps to steer the network into using information that is considered useful for making a prediction. Simultaneously, the amount of freedom for the network and regularizes the network optimization.

Thung and Wee [25] mention neurodegenerative disease diagnosis as an example of an application of multi-task learning. A model can be trained to predict multiple target outputs describing the disease progression at once. They state that learning to predict all these outputs together yields better performance than learning them separately, as these are related tasks. This idea can be applied to the prediction of Carnegie stages similarly. This can for instance be done by having the model predict Carnegie stage, gestational age and embryonic volume at once. This might steer the model in using more relevant information and have a regularizing effect.

4.4 Batch normalization

Batch normalization addresses the problem of internal covariate shift [26]. This shift occurs because the statistical characteristics of inputs to each layer vary during training when the network is trained in distinct batches. This slows down the training process since the layers need to constantly adapt to a new distribution of the input data. Batch normalization reduces this effect by fixing the means and variances of layer inputs, which aids in faster convergence and allows higher learning rates.

Ioffe and Szegedy [26] show that using batch normalization with a higher learning rate and without using dropout achieves the same performance with 5 times fewer training steps. Furthermore, they state that, when using batch normalization, dropout can be removed to speed up training, without increasing the risk of overfitting. Bjorck et al. [27] argue that using a higher learning rate increases the noise in a stochastic gradient descend step, which has a regularization effect. They argue that the regularization effect of batch normalization comes from the higher learning rates that it allows.

4.5 Weight initialization

Another way to implement regularization is by weight initialization. Weight initialization deals with choosing the best initial weights of the model before training. Most often, this is done by sampling the weights from a random distribution, such as a Gaussian distribution [12, 28]. It is also possible to pre-train the network for a different but similar task. An example of this is transfer learning.

Transfer learning is a method that is often used when limited data is available, in which a model is first trained on a larger amount of data that is similar to the target data [29, 30]. After this, the model is fine-tuned on the target data. This way, the model should already have learned useful features before training on the target domain [12]. The fine-tuning is most often done by fixing the first layers of the CNN (the feature extractor) and training the fully connected last layers on the target data. Several CNN models that were pre-trained with the ImageNet dataset, a dataset consisting of millions of natural images, are publicly available online.

4.6 Early stopping

Early stopping is a relatively simple method for regularization that prevents overfitting on the training set. Overfitting occurs when the error on the validation set starts to increase during training, while the error on the training set is still decreasing [20]. Early stopping halts the training process when this happens thereby reducing the generalization error.

5 Techniques for datasets with ordered classes

5.1 Multi-class classification and metric regression

The dataset for this project comprises 3D ultrasound scans with ordered Carnegie stage labels. If we approach this as a multi-class classification problem, we would treat the Carnegie stages as independent classes and use a loss function that doesn't penalize larger deviations between predicted and ground truth stages more. Because the inherent order in the labels is not used, this may lead the model to overlook meaningful relationships between classes and miss important characteristics for predicting the Carnegie stage [31].

Metric regression, on the other hand, works by having the model output a single continuous value. This way, it treats the labels as numerical values [32]. The optimization is then usually performed with a loss such as the mean squared error, which utilizes the order of the labels because larger deviations from the ground truth are punished more than smaller deviations. However, this approach can come with some issues when dealing with ordered labels. An example of this is in age estimation from facial images. Chang et al. [31] argue that using metric regression for age estimation from facial images is usually difficult since it is prone to overfitting. This is because the features that are related to the maturing of the face rely heavily on the person's age [32]. Facial shape is for instance an important feature to predict age for children, as the shape of the face changes a lot at a younger age. At an older age, on the other hand, this feature is less important since the facial shape does not change much during adulthood. Skin texture can tell a lot more about a person's age during this period, making this a more important feature. This change in the importance of features causes the feature space that describes age to be non-stationary, which is difficult to deal with for a regression model.

As described in section 3.3, the features that characterize the different Carnegie stages are quite non-stationary. Specifically, the features that are important for predicting earlier stages are very different from features that are important for predicting later stages. The orientation of the limbs is for instance very important for discerning stages 19 until 23, while this is less relevant for the earlier stages. Using metric regression might therefore cause similar issues as for age estimation.

In conclusion, neither multi-task classification nor metric regression seems fit for the task of embryonic Carnegie staging.

5.2 Deep ordinal regression

An often-used technique that overcomes the issues of multi-task learning and metric regression is ordinal regression. Ordinal regression generally learns how to predict labels with an ordinal scale, where only the relative order between labels is important [33]. Ordinal regression has been applied in numerous ways in machine learning, including for deep learning (deep ordinal regression).

This has been done in the field of age estimation. Niu et al. [32] propose a multiple-output CNN approach, with the same number of CNN outputs as the number of classes, which are called ranks, in the dataset minus one. Each output node is then trained with the binary classification task of predicting whether a sample has a higher rank than the one corresponding to that output node (output should be 1) or not (output should be 0). This means that the labels should be transformed into vectors with ones for each rank that the label surpasses and zeros when it doesn't. The output nodes are jointly trained with an absolute cost matrix that finds the absolute difference between prediction and ground truth for each node. Eventually, the model calculates the rank of an unseen sample by counting the number of nodes that are activated and adding one. Niu et al. [32] show that this ordinal regression CNN outperforms a metric regression CNN (with a single output node trained with the mean squared error) on two different facial image datasets.

A potential issue of this method, however, is the possibility of classifier inconsistency [34]. This happens when individual binary classifiers disagree, causing a contradiction. For example, when one classifier predicts that the stage of a subject is not higher than 20, while another predicts that the stage is higher than 21. In this case, there is no clear output of the model. Cao et al. [34] propose the consistent rank logits (CORAL) framework to solve this issue. This framework uses a weight-sharing constraint in the last layer so that all binary classification tasks share the same weight parameters.

This method, however, restricts the neural network's flexibility since the amount of trainable parameters is reduced. To overcome this, Shi et al. [35] introduce the conditional ordinal regression for neural networks (CORN) framework. Their framework uses an adapted training scheme with conditional training sets to ensure rank consistency. Instead of having each output node predict an unconditional probability, the output nodes predict the conditional probability of a rank being higher than the specific rank of that node, given that the rank is higher than that of the previous output node. The eventual unconditional probability is then found using the chain rule, by multiplying the conditional probability of this output node by the conditional probabilities of the lower-rank output nodes. To ensure that the output nodes render these conditional probabilities, an adapted training scheme is needed. This training scheme works by training the model output nodes separately, where each is trained on a subset of training data of which the ranks are at least larger than the rank of the output node below this node. Shi et al. [35] show that the CORN method does indeed have an improved performance over the CORAL method described by Cao et al. [34].

Another method that applies a form of ordinal regression for deep learning is described by Díaz and Marathe [36]. Their method uses soft ordinal vectors as labels instead of hard labels, such as one-hot encoded labels or rank labels as described in [32, 34, 35]. These labels are created by using a Softmax function that gives probability-like values to all ranks based on how far the rank of the sample is from each rank that corresponds to the element in the vector. This way, the highest value is given to the rank that corresponds to the sample and smaller values are given to ranks that are farther away. Hence, the order in the classes is implemented in the labels. The training procedure then follows that of a multi-class classification task, without the need for modification of the network architecture. The authors show that this method outperforms the ordinal regression CNN described by Niu et al. [32], however, it is unclear if this method works better than the CORAL [34] or CORN [35] methods.

6 Discussion

To the extent of our knowledge, this was the first project that attempted to develop a method for automated Carnegie staging of the human embryo based on 3D ultrasound. This literature study gave a broad overview of the literature on the most relevant topics for this project.

It was found that human raters mainly look at the limb development, curvature of the torso, and brain ventricle development for embryonic staging. It is expected that these features are also important for a neural network trained for this task. However, a neural network might also discover other features describing the Carnegie stages. This could be done by comparing the regions of the input scan that are important for the prediction of the model, using Grad-CAM [37] for instance, with the relevant features for that stage from literature. This can create insight into why the model might make a wrong prediction.

A variety of regularization techniques for deep learning classification tasks was discussed in section 4. According to the no-free-lunch theorem, however, it is not possible to come up with the best regularization strategy that works equally well for all classification tasks [8]. As such, an appropriate regularization strategy should be picked, based on the given task. This literature provides an overview of often-used methods to help a model generalize better to unseen data when only limited data is available.

As a starting point, Moradi et al. [8] recommend the use of weight decay and data augmentation for regularization. Furthermore, they recommend dropout or batch normalization in case there are enough computational resources. Kukacka et al. [12] recommend using data augmentation that mimics natural transformations in the data. Furthermore, they recommend experimenting with out pre-trained or random weight initializations and different optimizers and learning rates. Additionally, Nusrat and Jang [13] have shown that batch normalization and data augmentation lead to better performance than L_1 regularization. Ioffe and Szegedy [26] state that batch normalization reduces the need for dropout as it provides similar benefits as dropout.

Based on these recommendations, data augmentation and batch normalization seem like the most promising regularization techniques for this project. Additionally, multi-task learning could be used to have the model predict both the Carnegie stage and other relevant embryonic variables at the same time. This could help steer the model in using more relevant information and thereby improve generalization. Furthermore, different optimizers, learning rates, and weight initializations could be tried.

For the case of deep learning applications, Shi et al. [35] have shown that their framework with conditional training sets solves the rank inconsistency problem and the reduced model capacity problem, introduced by Cao et al. [34]. It is therefore recommended to use this method. However, it is unclear if this method works better than the soft ordinal vector labels method by Díaz and Marathe [36]. Therefore, it is recommended to try both methods.

7 Conclusion

The most relevant embryonic features for Carnegie staging in 3D ultrasound by human raters are limb development, trunk curvature, and brain ventricle development. Data augmentation and batch normalization seem like the most promising regularization techniques for this project. Furthermore, it is recommended to implement a deep ordinal regression framework, either with conditional training sets or with soft ordinal vectors as labels.

References

[1] Ronan O'Rahilly and Fabiola Müller. Developmental Stages in Human Embryos: Revised and New Measurements. *Cells Tissues Organs*, 192(2):73–84, July 2010.

[2] Christine M. Verwoerd-Dikkeboom, Anton H. J. Koning, Peter J. van der Spek, Niek Exalto, and Eric A. P. Steegers. Embryonic staging using a 3D virtual reality system. *Human Reproduction*, 23(7):1479–1484, July 2008.

[3] Mark A. Hill. Early human development. *Clinical obstetrics and gynecology*, 50(1):2–9, 2007.

[4] Regine P.M. Steegers-Theunissen, Jennifer J.F.M. Verheijden-Paulissen, Evelyne M. van Uitert, Mark F. Wildhagen, Niek Exalto, Anton H.J. Koning, Alex J. Eggink, Johannes J. Duvekot, Joop S.E. Laven, Dick Tibboel, et al. Cohort profile: the rotterdam periconceptional cohort (predict study). *International Journal of Epidemiology*, 45(2):374–381, 2016.

[5] Melek Rousian, Sam Schoenmakers, Alex J. Eggink, Dionne V. Gootjes, Anton H.J. Koning, Maria P.H. Koster, Annemarie G.M.G.J. Mulders, Esther B. Baart, Irwin K.M. Reiss, Joop S.E. Laven, et al. Cohort profile update: the rotterdam periconceptional cohort and embryonic and fetal measurements using 3d ultrasound and virtual reality techniques. *International Journal of Epidemiology*, 50(5):1426–1427l, 2021.

[6] Melek Rousian, Wim C.J. Hop, Anton H.J. Koning, Peter J. Van der Spek, Niek Exalto, and Eric A.P. Steegers. First trimester brain ventricle fluid and embryonic volumes measured by three-dimensional ultrasound with the use of i-space virtual reality. *Human Reproduction*, 28(5):1181–1189, 2013.

[7] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical Image Analysis using Convolutional Neural Networks: A Review. *Journal of Medical Systems*, 42(11):1–13, November 2018.

[8] Reza Moradi, Reza Berangi, and Behrouz Minaei. A survey of regularization strategies for deep models. *Artificial Intelligence Review*, 53(6):3947–3986, August 2020.

[9] Ronan O'rahilly, Fabiola Muller, and George L. Streeter. Developmental stages in human embryos: including a revision of streeter's horizons and a survey of the carnegie collection. 1987.

[10] Sander Flierman, Melanie Tijsterman, Melek Rousian, and Bernadette S. de Bakker. Discrepancies in Embryonic Staging: Towards a Gold Standard. *Life*, 13(5):1084, April 2023.

[11] Samir S. Yadav and Shivajirao M. Jadhav. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, 6(1):1–18, December 2019.

[12] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for Deep Learning: A Taxonomy. *arXiv*, October 2017.

[13] Ismoilov Nusrat and Sung-Bong Jang. A Comparison of Regularization Techniques in Deep Neural Networks. *Symmetry*, 10(11):648, November 2018.

[14] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, August 2021.

[15] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):1–48, December 2019.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, October 2020.

[17] Luis Perez and Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv*, December 2017.

[18] Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. Model complexity of deep learning: a survey. *Knowl. Inf. Syst.*, 63(10):2585–2619, October 2021.

[19] Lorenzo Brigato and Luca Iocchi. *A Close Look at Deep Learning with Small Data*. IEEE Computer Society, January 2021.

[20] Ekaba Bisong. Regularization for deep learning. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pages 415–421, 2019.

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[22] Pierre Baldi and Peter J Sadowski. Understanding dropout. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[23] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *ArXiv*, abs/1706.05098, 2017.

[25] Kim-Han Thung and Chong-Yaw Wee. A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22):29705–29725, November 2018.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[27] Nils Bjorck, Carla P. Gomes, Bart Selman, and Kilian Q. Weinberger. Understanding Batch Normalization. *Advances in Neural Information Processing Systems*, 31, 2018.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[29] Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C. S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong,

Made K. Prasadha, Jacqueline Pei, Magdalene Y. L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A. N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5):1122–1131.e9, February 2018.

[30] Hee E. Kim, Alejandro Cosa-Linan, Nandhini Santhanam, Mahboubeh Jannesari, Mate E. Maros, and Thomas Ganslandt. Transfer learning for medical image classification: a literature review. *BMC Medical Imaging*, 22(1):1–13, December 2022.

[31] Kuang-Yu Chang, Chu-Song Chen, and Yi-Ping Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *CVPR 2011*, pages 20–25. IEEE.

[32] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal Regression with Multiple Output CNN for Age Estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27–30. IEEE.

[33] Huan Fu, Mingming Gong, Chaohui Wang, K. Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[34] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331, December 2020.

[35] Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955, August 2023.

[36] Raúl Díaz and Amit Marathe. Soft labels for ordinal regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4742, 2019.

[37] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.