# Fracture Risk Prediction Using Multimodal Neural Networks

## A Study on the Effect of Incorporating Spinal Radiographs into Multimodal Neural Networks for Osteoporotic Fracture Risk Assessment

Master's thesis in Biomedical Engineering

ERIK BLOMQVIST

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

# Fracture Risk Prediction Using Multimodal Neural Networks

A Study on the Effect of Incorporating Spinal Radiographs into Multimodal Neural Networks for Osteoporotic Fracture Risk Assessment

ERIK BLOMQVIST


CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
*Computer Vision and Medical Image Analysis*
Chalmers University of Technology
Gothenburg, Sweden 2025

Fracture Risk Prediction Using Mulrimodal Neural Networks
A Study on the Effect of Incorporating Spinal Radiographs into Multimodal Neural Networks for Osteoporotic Fracture Risk Assessment
ERIK BLOMQVIST



Supervisor: Victor Wåhlstrand, Department of Electrical Engineering
Examiner: Jennifer Alvén, Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Computer Vision and Medical Image Analysis
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Spinal radiograph from the SUPERB dataset used for fracture risk prediction.

Typeset in LaTeX
Gothenburg, Sweden 2025

Fracture Risk Prediction Using Multimodal Neural Networks
A Study on the Effect of Incorporating Spinal Radiographs into Multimodal Neural Networks for Osteoporotic Fracture Risk Assessment
ERIK BLOMQVIST
Department of Electrical Engineering
Chalmers University of Technology

# Abstract

Osteoporosis, characterized by reduced bone mass and micro-architectural deterioration, increases the risk of fractures, particularly in elderly populations. Traditional fracture risk prediction models rely on clinical risk factors but do not incorporate imaging data, potentially overlooking structural indicators. This thesis explores the potential of multimodal learning by integrating spinal X-ray images with clinical risk factors to improve fracture risk prediction. Both full spinal radiographs and cropped vertebral images are evaluated to investigate whether focusing on anatomically relevant regions can enhance the predictive signal. Two deep learning architectures are considered, Convolutional neural networks (CNNs) and Vision transformers (ViTs), alongside multiple fusion strategies for combining image and tabular data. Results demonstrate that multimodal models consistently outperform baselines, with the best performance achieved by a CNN using vertebral crops and intermediate fusion (C-index: 0.69, AUC: 0.76, Brier score: 0.14). This suggests that image data alone contain meaningful predictive information, and that combining imaging with clinical features enhances fracture risk prediction. Using vertebral crops as input generally yielded better performance than using full radiographs, highlighting the importance of localized features. However, the models were evaluated on a single dataset of elderly Caucasian women, indicating the need for future work to assess generalization across diverse populations.

# List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

| | |
|---|---|
| APR | Anterior-posterior ratio |
| AUC | Area under the curve |
| BERT | Bidirectional Encoder Representations from Transformers |
| BS | Brier score |
| C-DS | ConvDeepSurv |
| C-index | Concordance index |
| CNN | Convolutional neural network |
| CPH | Cox proportional hazards |
| FPR | False positive rate |
| GSQ | Genant's semi-quantitative method |
| HPR | Posterior-posterior ratio |
| IF-MLP | Intermediate fusion multi layer perceptron |
| IPCW | Inverse probability of censoring weighting |
| KM | Kaplan-Meier |
| LF-FC | Late fusion fully connected |
| LN | Layer normalization |
| MLP | Multi layer perceptron |
| MPR | Middle-posterior ratio |
| MSA | Multi-head self-attention |
| NLP | Natural language processing |
| NN | Neural network |
| ReLU | Rectified linear unit |
| ROC | Reciever operating characteristic |
| SA | Self-attention |
| SUPERB | Sahlgrenska University Hospital Prospective Evaluation of Risk of Bone Fractures |
| TPR | True positive rate |
| ViT | Vision transformer |
| ViT-DS | ViTDeepSurv |
| XVFA | Explainable vertebral fracture analysis |

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

Osteoporosis is a medical condition marked by the loss of bone mass and the deterioration of bone micro-architecture, leading to an increased risk of fractures [1]. It is most prevalent among the elderly, especially among women of Caucasian descent [2]. Osteoporotic fractures are associated with increased morbidity, reduced quality of life, and higher mortality rates [3]. Despite its significant health impact, osteoporosis often goes undiagnosed until one or more fractures have already occurred. This highlights the importance of early identification of individuals at high risk, enabling intervention and treatment to prevent fractures.

Fracture risk prediction models are essential tools in clinical practice to support decision making regarding osteoporosis management. Traditional models such as FRAX® [4], rely on clinical risk factors like age, prior fracture history and bone mineral density to estimate the 10-year probability of fractures [5]. Such models are valuable for fracture risk prediction, but they do not leverage all available information. Medical imaging data such as X-ray scans of the bone could help capture subtle structural changes and potentially carry predictive information of fractures, but images cannot be incorporated in traditional models directly.

Medical imaging may reveal structural abnormalities or vertebral fractures that are not always clinically recognized. Several studies have explored the use of such imaging data to assess fracture risk more accurately [6], [7]. However, these imaging features are typically underutilized in clinical risk scores due to challenges in quantifying them consistently and integrating them into prediction models.

Deep learning has grown rapidly in recent years and show great promise in medical imaging tasks, especially due to the ability to learn complex patterns from data [8]. Convolutional Neural Networks (CNNs) have been widely used for tasks based on imaging due to their ability to detect spatial relationships. In particular, convolutional neural networks have shown strong performance in vertebral segmentation on computed tomography and magnetic resonance imaging scans [9]. Additionally, they have been proved to be able to classify fracture risk patients using radiographs [10]. Vision transformers (ViTs) is another architecture that has been developed for image classification tasks more recently, and promising results have been achieved, especially in scenarios with complex spatial dependencies [11], [12]. Vision transformers use self-attention mechanisms originally developed for natural language processing tasks to help capture long-range dependencies within images, which could be important for medical imaging contexts.

Multimodal learning refers to machine learning approaches that incorporate multiple data modalities for learning certain patterns [13]. Such data can be clinical variables and medical images, and this paradigm has grown due to its ability to combine complementary information - clinical variables may provide one type of information while medical images may provide another. Recent studies have shown that multimodal deep learning models can outperform unimodal models in various tasks, such as skin cancer detection and Alzheimer's disease detection [14], [15].

In this project, spinal X-ray images and clinical risk factors are used to evaluate the effectiveness of multimodal neural networks in comparison to unimodal models for fracture risk prediction. Two strategies for incorporating imaging data are explored. Using entire spinal X-rays, and using extracted vertebral crops in an attempt to enhance signal. Additionally, two deep learning architecures are compared to evaluate their effectiveness in this context, convolutional neural networks and vision transformers.

## 1.1 Aim

The dataset used in this project is the Sahlgrenska University Hospital Prospective Evaluation of Risk of Bone Fractures. It consists of spinal X-ray images, clinical risk factors and fracture outcomes from a population of older Caucasian women. This project aims to address the gap in current fracture risk prediction models such as FRAX® by exploring the effectiveness of multimodal neural networks that integrate both clinical risk factors and spinal X-ray images.

Two different approaches of incorporating images will be explored in an attempt to achieve better predictive performance. The first is by using entire spinal X-ray images as input to multimodal neural networks to capture global information on patient-level. The second is by using crops of individual vertebrae as input to the models in an attempt to reduce background noise and focus on relevant information. Crops from each individual subject will be concatenated and used as one input, such that the prediction remain on patient-level.

It is hypothesized that multimodal models—combining image and tabular inputs—will outperform models trained solely on clinical risk factors, by capturing complementary signals from both data types. Furthermore, it is expected that using localized image regions focused on individual vertebrae, rather than entire spinal images, will lead to improved predictive performance. This is based on the assumption that vertebral crops emphasize informative structures while reducing irrelevant background and noise.

In addition to comparing these approaches, the project aims to investigate whether one neural network architecture is preferable for fracture risk prediction. The developed architectures will be evaluated in order to determine their effectiveness in integrating multimodal data for this purpose. The desired outcome for this project will be demonstrating whether multimodal models outperform models based solely on clinical risk factors, as well as identifying potential benefits of specific architectures and image input strategies for fracture risk assessment.

## 1.2 Scope

The scope of the project is limited by time, and consequently by the number of models to be evaluated. The comparison is restricted to two overarching architectures - convolutional neural networks and vision transformers. Moreover, the project focuses exclusively on neural network based survival models with DeepSurv as backbone as opposed to other types of statistical methods. Prediction is performed either using tabular data alone, images alone, or multimodal combinations of both, using intermediate or late fusion strategies. Other fusion strategies are not considered in this project.

Most importantly, the project is limited to the dataset being used. As all patients in the cohort are women, specifically older Caucasian women, the outcomes of this project will only be applicable to that part of the population. In order to yield more transferable results, other datasets would be needed from other subgroups, which are not available. The data makes the distinction of general fracture as any fracture not considered hip fracture or major osteoporotic fracture. While the dataset also includes information regarding hip fractures and major osteoporotic fractures, the study is limited to only analyze the occurrence of general fractures. The reason is that substantially fewer events are present in the data for those specific fractures, and it is expected that using them as target events would cause model instabilities and poor generalization.

# 2

# Background

## 2.1 Data

The Sahlgrenska University Hospital Prospective Evaluation of Risk of Bone Fractures (SUPERB) [1] is a cohort of 3028 Swedish women aged 77.8±1.6, with recorded risk factors, low-dose spinal X-rays and fracture outcomes for the hip, general fractures and so-called *major osteoporotic fractures*. The dataset also includes computed tomography volumes collected at four different locations for each patient. The criterion for defining fractures is Genant's semi-quantitative method (GSQ) [16]. In this method, six keypoints are annotated on each vertebra in a spinal X-ray image, and both morphology and severity of fractures are classified. The keypoints define the anterior, posterior and middle vertebral heights, and ratios between anterior-posterior (APR), middle-posterior (MPR) and posterior-posterior (HPR) are calculated. A prevalent fracture is then defined as a 15% difference in APR, MPR or HPR compared to the mean value of a normal population, see Figure 2.1.

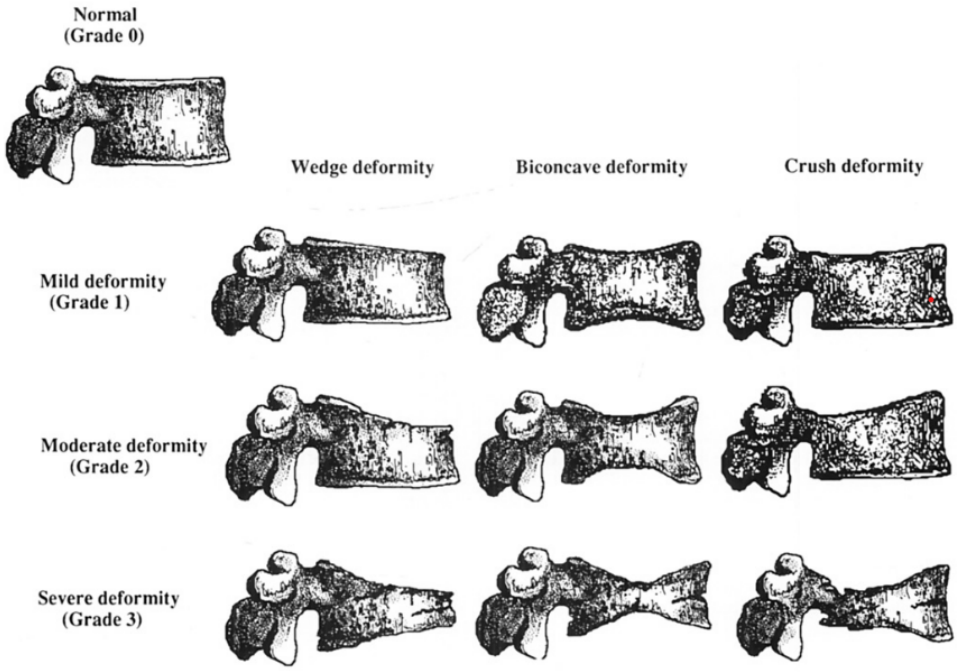



**Figure 2.1:** Semiquantitative visual grading of vertebral deformities, from [16].

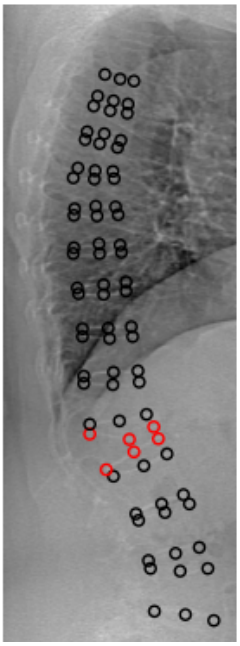
**Table 2.1:** Summary statistics of outcome variables and clinical risk factors in the SUPERB dataset. Binary variables are reported as counts and percentages, continuous variables are shown as mean ± standard deviation.

| Variable description | Summary statistics |
|---|---|
| Any fracture experienced | 1083 (35.8%) |
| Time to any fracture, days | 2317.0 ± 991.8 |
| Age, years | 77.8 ± 1.6 |
| BMI, kg/m$^3$ | 26.3 ± 4.4 |
| Prior fracture | 1117 (36.9%) |
| Parental history of hip fracture | 533 (17.6%) |
| Current smoker | 158 (5.2%) |
| Glucocorticoid steroid usage | 103 (3.4%) |
| Rheumatoid arthritis | 120 (4.0%) |
| Excessive alcohol consumption | 17 (0.6%) |
| Secondary osteoporosis | 787 (26.0%) |
| Normalized bone mineral density score | -1.6 ± 0.9 |

The outcome variables and clinical risk factors summarized in Table 2.1 are used as input features in the fracture risk prediction models. More detailed descriptions of the variables are available in [1].

The spinal X-ray images in SUPERB are large and of varying size, around 1200×600 pixels to 1600×600 pixels. The majority of pixels depict what could be considered background - including fat and other soft tissues, whereas the spine is the region of interest. A potential solution to this is to use cropped parts of the image, depicting only the vertebrae themselves, described further in Section 3.1.6.

**(a)** Example of a spinal X-ray image from a patient who did not have a fracture at the start of the study.



**(b)** Example of a spinal X-ray image from a patient who did have a fracture at the start of the study.

**Figure 2.2:** Comparison of spinal X-ray images for patients with and without fractures.

The sample X-rays in Figures 2.2a and 2.2b depict two different patients who did not have a fracture and who did have a fracture respectively. In Figure 2.2b, keypoints of individual vertebrae are marked using GSQ, with the red being the vertebra with a fracture. While the vertebrae are identifiable, the image does not contain any obvious radiological signs of a future fracture, other than the general condition of the patient.

## 2.2 Survival Analysis

Survival analysis is a field within statistics focused on analyzing the time until an event occurs [17]. The name survival analysis comes from the fact that the event commonly analyzed is death, but it could just as well be disease incidence, recovery, machine failure, or as in the case of this thesis, bone fracture. The time is measured from a defined starting point, usually from the beginning of follow-up of an individual, until the occurrence of an event.

As opposed to standard regression techniques, survival analysis requires the ability to handle censored data, where the outcome is only partially observed. This property is essential in many real-world studies where not all individuals experience the event during the observation period.

### 2.2.1 Survival and Hazard Functions

The survival function is an essential part of survival analysis and gives the probability of an individual surviving, meaning they have not experienced an event, past a specified time $t$. It can be expressed according to Equation 2.1,

$$S(t) = P(T > t), \tag{2.1}$$

where $T$ is a random variable denoting the time of the event. The survival function is always non-increasing, starting as $S(0) = 1$ since all subjects are event-free at the start.

A closely related function is the hazard function, $h(t)$. It is used to calculate the instantaneous potential per time unit for an event to occur, given that the patient has not already experienced an event up until $t$ [17]. This can be denoted by the limit

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t}. \tag{2.2}$$

In other words, the hazard function describes the rate at which the subjects are experiencing the event at a given time point $t$. The hazard function is always non-negative and has no upper bound.

### 2.2.2 Censoring

A challenge in survival analysis is censored data, which refers to entries with incomplete event time observations [17]. The most common form of censoring is called right-censoring, which means that the survival time is observed up until a certain time point, but not after. This happens, for example, if patients withdraw from the study or simply do not experience the event before the study ends. Censoring of a patient $i$ at time $T_i$ may be denoted by a binary indicator random variable $\delta_i$, where $\delta_i = 1$ denotes an event occurring at time $T_i$, and $\delta_i = 0$ that the patient was censored at that time. As censoring is not unusual in real-world data, it is essential that survival models have the ability to handle it.

### 2.2.3 Kaplan-Meier Estimate

The Kaplan-Meier (KM) estimate is a way to measure the fraction of subjects not having experienced a specified event for a certain amount of time after treatment or start of study [18]. Effectively, it estimates the survival function of a population when calculated at multiple time points. KM is effective at computing the survival over time even when censored data is present. The estimation is made by computing probabilities of event occurrences at certain time points, and then multiplying them with the computed probabilities from earlier time points. At each time point, the probability is calculated by dividing the number of surviving subjects (i.e. subjects who have not yet experienced the event) by the number of subjects at risk. For a time point $t$, this yields Equation 2.3:

$$\hat{S}(t) = \prod_{t' \leq t} \left(1 - \frac{n(t')}{R(t')}\right) \tag{2.3}$$

where $n(t')$ is the number of events at time $t'$, and $R(t')$ is the number of subjects at risk just before $t'$. The risk set $R(t')$ includes all subjects still under observation and event-free immediately before $t'$.

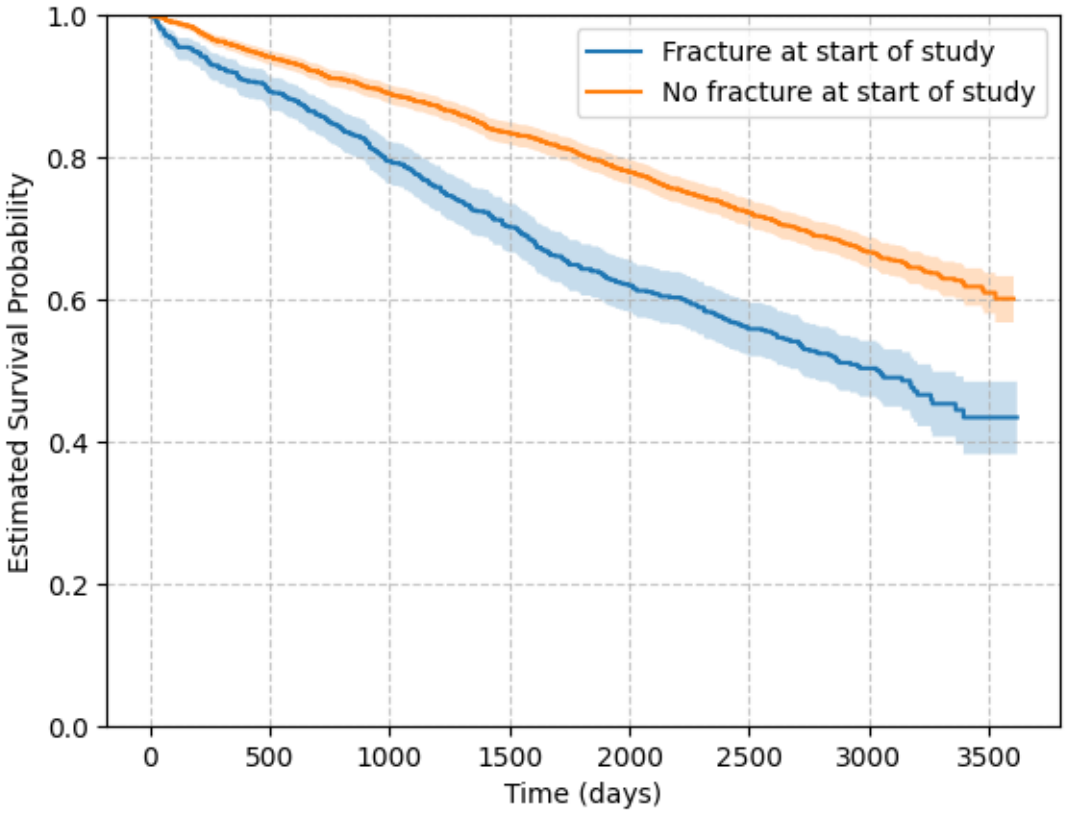



**Figure 2.3:** Kaplan-Meier estimate of survival functions (event any fracture) of two subgroups of the population in the SUPERB dataset, with 95% confidence interval.

Figure 2.3 exemplifies two KM estimates of the SUPERB dataset. As can be seen, they are steadily decreasing since start of study, although having a fracture at start of study yields lower chances of being free of upcoming fractures according to the estimate.

### 2.2.4 Cox Proportional Hazards

The Cox proportional hazards (CPH) model is used in survival analysis to analyze the relationship between hazard and input variables and allows for the inclusion of covariates in the analysis [17]. The model assumes that the hazard at any time is a product of a baseline hazard function $h_0(t)$ and a time-independent exponential term which depends on the input variables, $\mathbf{X}$. The resulting hazard function is expressed in Equation 2.4:

$$h(t, \mathbf{X}) = h_0(t)e^{\sum_{i=1}^{p} \beta_i X_i}. \tag{2.4}$$

Here, $\beta_i$ denotes the model coefficient for the $i$th input parameter, $X_i$, and represents the log of the hazard ratio associated with a one unit increase in $X_i$. An important

property of the CPH model is that the hazard ratios between any two subjects remain constant over time [17]. For the inputs $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, the hazard ratio is

$$\frac{h(t, \mathbf{X}^{(1)})}{h(t, \mathbf{X}^{(2)})} = \frac{h_0(t)e^{\boldsymbol{\beta}^\top \mathbf{X}^{(1)}}}{h_0(t)e^{\boldsymbol{\beta}^\top \mathbf{X}^{(2)}}} = e^{\boldsymbol{\beta}^\top (\mathbf{X}^{(1)} - \mathbf{X}^{(2)})}, \tag{2.5}$$

which is not dependent on time $t$. This proportional hazards assumption means that the relative risk between subjects is time invariant. In other words, the effect of a covariate is multiplicative with respect to the hazard and does not change with time. Since the model is log-linear, the log of the hazard ratio between subjects is a linear function of the difference in their covariates. Importantly, it enables the model to be used to analyze long-term effects of input variables without having to define the baseline hazard function explicitly [17].

### 2.2.5 Loss Function

In order to optimize the risk predictions of the models, a loss function need to be defined. For survival analysis models, Cox partial likelihood is well-suited, particularly because it handles censored data effectively [19]. The partial likelihood is derived from the CPH model and estimates the relative risk without needing to model the baseline hazard $h_0(t)$ due to the proportional hazards assumption. It is only partial since it focuses on the ordering of events rather than exact event times. The idea of the loss function is to assign higher predicted risk scores to subjects who experience the event earlier.

The model yields a risk score $\hat{r}_i = \boldsymbol{\beta}^\top \mathbf{X}_i$ for each subject $i$, where higher scores indicate greater predicted risk. For subject $i$ with event time $t_i$ and event indicator $\delta_i$, the partial likelihood compares this subject's risk to those still at risk at time $t_i$. The set of those still at risk is defined as the risk set,

$$R(t_i) = \{j \mid t_j \geq t_i\},$$

i.e., all individuals who have not yet experienced an event or been censored at $t_i$. The loss $\mathcal{L}_{Cox}$ is then defined as the negative log partial likelihood:

$$\mathcal{L}_{Cox} = -\sum_{i=1}^{n} \delta_i \left[ \hat{r}_i - \log \left( \sum_{j \in R(t_i)} \exp(\hat{r}_j) \right) \right], \tag{2.6}$$

where $\hat{r}_i$ is the predicted risk score for subject $i$, $\delta_i$ the event indicator and $R(t_i)$ is the risk set at time $t_i$. By this design, the loss encourages the model to give higher risk scores to subjects who experience the event earlier, compared to those still at risk. Censored subjects contribute only to the risk sets of earlier events and not to the numerator, which allows the model to learn from censored data as well.

In implementation, this loss can be computed using cumulative sums after sorting subjects by descending event time. A small constant is added to avoid numerical instability in the logarithm.

## 2.3 Neural Networks

Artificial neural networks (NN) are a class of machine learning models that originally tried to imitate the brain's neural network processing [20]. They consists of layers of neurons, interconnected nodes, where each connection has an associated weight and bias. These types of models are used for learning complex, nonlinear relationships in data.

In feedforward neural networks, data flows in one direction, from the input layer through hidden layers to the output layer. The pass of data through each layer is computed by a weighted sum of the layer's inputs

$$z = \sum_i w_i x_i + b \tag{2.7}$$

where $x_i$ is the $i$th input feature, $w_i$ its corresponding weight, and $b$ is the bias term.

In turn, each node represents a specific output function, called activation function. In order for networks to learn nonlinear behaviors, activation functions such as the rectified linear unit (ReLU) is used. ReLU is defined as

$$a = f(z) = \max(0, z), \tag{2.8}$$

which ensures that only positive values pass through, introducing nonlinearity. Without nonlinearity, the network would be unable to model complex patterns in data which is important since real world data rarely follow purely linear relationships.

Training of artificial neural networks is typically done using backpropagation together with gradient descent to minimize a loss function $\mathcal{L}$ that quantifies the difference (error) between model predictions and true values. Backpropagation is an algorithm that computes the gradient of the loss function with respect to the different weights in the network through the chain rule. The error is propagated backwards through the model, updating each weight in proportion to how much the weight contributed to the error. Given a weight $w$, the update rule is

$$w \leftarrow w - \eta \frac{\partial \mathcal{L}}{\partial w}, \tag{2.9}$$

where $\eta$ is the learning rate and $\frac{\partial \mathcal{L}}{\partial w}$ is the partial derivative of the loss function with respect to the weight. This process is performed iteratively over many passes of data (epochs), such that the weights are refined to minimize $\mathcal{L}$, ideally converging to a solution that generalizes well to unseen data. For prediction tasks, common loss functions include cross-entropy and mean squared error for classification and regression respectively.

### 2.3.1 Convolutional Neural Networks

The first convolutional neural network (CNN) was introduced in 1989 by LeCun et al. [21] in their work on handwritten digit recognition, and later refined into

the well-known LeNet-5 architecture in 1998 [22]. The mathematical foundations of CNNs had however been established earlier. CNNs generally consist of multiple layers that apply mathematical operations to extract hierarchical features from the input. These layers typically include convolutional layers, followed by activation functions and pooling layers, grouped into repeated blocks.

The convolutional layer is the core of the CNN which applies the convolution operation to its input. The input is usually a multi-dimensional matrix, either the original input for the first layer or feature maps for the subsequent layers. In the layer, a different filter kernel is applied independently to each input channel, which means that 2D convolution is performed. By defining the convolution operation as $*$, the general form of the operation in CNNs can be written as

$$y_{c'} = \sum_{c} x_c * w_{c',c}, \tag{2.10}$$

where $c$ denotes channel, $x_c$ is the input feature map, $w_{c',c}$ is the kernel with filter weights, and $y_{c'}$ is the output feature map. More explicitly, this can be expressed as

$$y_{i,j,c'} = \sum_{c} \sum_{m} \sum_{n} x_{i-m,j-n,c} w_{m,n,c,c'}, \tag{2.11}$$

where $x_{i-m,j-n,c}$ is the input feature at position $(i-m, j-n)$ in channel $c$, and $w_{m,n,c,c'}$ are the weights of the kernel connecting input channel $c$ to output channel $c'$. The output feature map $y_{i,j,c'}$ is computed by summing over all input channels and kernel positions.

Evidently, the convolutional layers make use of weight sharing, i.e. the same filter is applied across the entire spatial dimension of the input, which significantly reduces the number of learnable parameters of the network. Two other aspects of the convolutional layers are stride and padding. Stride controls how much the filter moves across the input in each step, where $S = 1$ means one pixel at a time, which preserves more spatial details. By using $S > 1$ the filter moves multiple pixels at a time, preserving less spatial details and reducing the output size. Padding is also used to control the size of spatial features by adding additional pixels outside the borders of the input, commonly with zeros. By using padding, $P > 0$, one can ensure that the output size is kept equal to the input size. With no padding, $P = 0$, the output size will decrease in the same way as in normal convolution. The output size considering stride and padding can be expressed according to Equation 2.12,

$$H' = \frac{H - F_H + 2P}{S} + 1, \qquad W' = \frac{W - F_W + 2P}{S} + 1, \tag{2.12}$$

where $H, W$ is the height and width of the input $x$, $F_H, F_W$ the height and width of the filter kernel $w$ and $H', W'$ the height and width of the output $y$.

After convolution, a nonlinear activation function is applied element-wise to introduce nonlinearity, most commonly ReLU. The final step of the block is the pooling layer, which purpose is to reduce the spatial dimensions while still retaining essential information. During pooling, the input is divided into smaller subgroups, where only one value is saved from each. Most commonly max pooling is used, which

simply saves the maximum pixel value in each subgroup. This can be expressed by Equation 2.13

$$x'_{i,j} = \text{pool}(z_{i,j}) = \max_{(p,q)\in\mathcal{R}} z_{i+p,j+q}, \tag{2.13}$$

where $\mathcal{R}$ is the pooling region and $x'$ is the output of the layer, and subsequently the input to the next convolutional layer. By chaining these layers into a block we can simply define the input to the next block of layers according to Equation

$$x' = \text{pool}(f(x * w + b)), \tag{2.14}$$

where $b$ is an added bias term.

### 2.3.2 Vision Transformers

Vision transformers (ViTs) are inspired by the Transformer architecture, which was originally developed for natural language processing (NLP) tasks [23]. The first ViT model for image classification was introduced by Dosovitskiy et al. [11], adapting the standard Transformer architecture with minimal modifications to process image data effectively. Unlike Convolutional neural networks, which inherently capture local spatial patterns through convolutional filters, ViTs rely on self-attention mechanisms to model global dependencies between image regions.

For patch embedding in ViTs, an image $x \in \mathbb{R}^{H\times W\times C}$, where $x_{i,j,c}$ denotes the input at height $i$, width $j$ and channel $c$, is first divided into smaller patches of size $P \times P$. The total number of patches is given by:

$$N = \frac{HW}{P^2}. \tag{2.15}$$

Each patch is flattened into a vector of size $P^2C$ and projected into a $D$-dimensional embedding space using a learnable embedding matrix. For $n \in \{1, ..., N\}$ the patch embedding for the $n$th patch then becomes

$$z_0^n = W_E x_n + b_E, \tag{2.16}$$

where $W_E \in \mathbb{R}^{(P^2C)\times D}$ is the learnable embedding matrix, $x_n$ the flattened patch vector and $b_E$ a bias term. In order to preserve spatial information, learnable positional embeddings $E_{\text{pos}} \in \mathbb{R}^{(N+1)\times D}$ are added to the patch embeddings, and they can be expressed as

$$Z_0 = [z_{\text{class}}; z_0^1; z_0^2; ...; z_0^N] + E_{\text{pos}}, \tag{2.17}$$

where $z_{\text{class}}$ is the classification token, which, similarly to how the classification token works in Bidirectional Encoder Representations from Transformers (BERT) [24] for NLP tasks, serves as a global representation of the image.

The sequence of patch embeddings $Z_0$ is then processed by a standard transformer encoder, consisting of multiple layers of Multi-Head Self-Attention (MSA) and Feed-Forward Networks. The self-attention (SA) mechanism computes attention scores between all patches to determine how much each patch should influence the other

patches. The self-attention computes queries $Q$, keys $K$ and values $V$ according to Equation 2.18,

$$Q = ZW_Q, \qquad K = ZW_K, \qquad V = ZW_V, \tag{2.18}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{D \times D}$ are learnable weight matrices. The attention scores $\mathrm{SA}(Z)$ can then be computed using scaled dot-product attention,

$$\mathrm{SA}(Z) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{D}}\right)V, \tag{2.19}$$

where softmax$(\cdot)$ is an activation function that restricts its output between 0-1, essentially weighting the values $V$ to compute the attention scores. MSA is then an extension of SA where $k$ heads perform the operation in Equation 2.19 in parallel, resulting in Equation 2.20

$$\mathrm{MSA}(Z) = [\mathrm{SA}_1(Z); \mathrm{SA}_2(Z); ...; \mathrm{SA}_k(Z)]W_{\mathrm{MSA}}. \tag{2.20}$$

Here, $k$ denotes the number of heads and $W_{\mathrm{MSA}}$ is a learnable output projection matrix. The output of the MSA is then fed through layer normalization (LN) and a multi-layer perceptron (MLP), resulting in Equation 2.21:

$$Z' = \mathrm{MLP}(\mathrm{LN}(Z + \mathrm{MSA}(Z))). \tag{2.21}$$

This is repeated for multiple transformer encoder blocks to extract higher level features from the input image. These transformer encoder blocks are finally followed by an MLP head for final prediction.

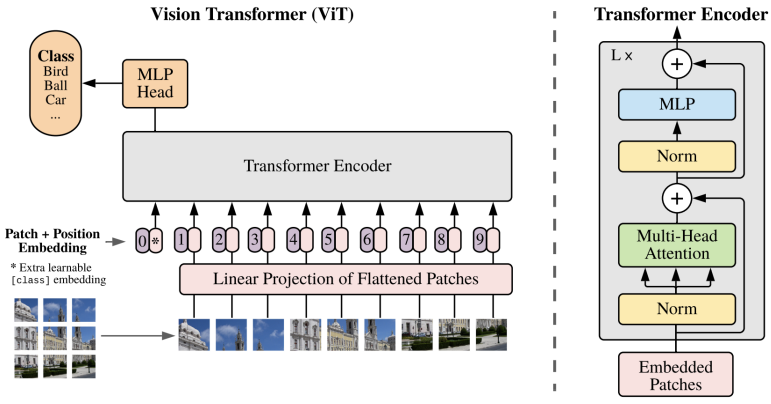



**Figure 2.4:** Overview of the vision transformer architecture from [11]. Image patches are flattened, embedded and fed through a transformer encoder with multi-head self-attention.

Figure 2.4 shows an overview of ViT, with patch embedding and prediction on the left and $L$ blocks of transformers encoders on the right.

## 2.4 DeepSurv

DeepSurv is a CPH deep neural network survival method, designed to handle complex, non-linear relationships in survival data [19]. The model calculates the effects of observed covariates on the risk of an event occurring, typically by estimating log-hazards. The hazard can be defined

$$h(t, \mathbf{X}) = h_0(t)e^{\phi(\mathbf{X})} \tag{2.22}$$

which is similar to the Cox Proportional hazard in Equation 2.4, with the key difference that $\phi(\mathbf{X})$ is modeled by a fully connected multi-layer neural network and has no linear restrictions - the normal CPH model assumes that a patient's log hazard of event is a linear combination of their covariates, while DeepSurv models the relationships as non-linear.

A straightforward generalization of DeepSurv is to allow $\phi(\mathbf{X})$ to be modeled by architectures other than fully connected networks. As demonstrated in DeepConvSurv, convolutional layers can be used to process image inputs [25].

## 2.5 Explainable Vertebral Fracture Analysis

Explainable vertebral fracture analysis (XVFA) is a method for vertebral fracture assessment using deep neural networks [26]. The assessment incorporates vertebra detection and keypoint localization with uncertainty estimates as well as vertebra fracture grade and morphology.

Vertebrae are detected through a two-stage method of coarse bounding box detection using a transformer architecture. Fracture morphology varies between normal, concave, wedge and crush deformities, and severity vary between normal, mild, moderate and severe. Both morphology and severity are classified using Genant's semi-quantitative method, which is a differentiable and rule-based means of classification. By using rule-based classification, the decisions of XVFA are interpretable.

In evaluations, XVFA achieved a vertebra-level sensitivity of 93% and an end-to-end area under the reciever operating characteristic curve of 97% on low-dose spinal radiographs. These results demonstrate its effectiveness in both detecting vertebrae and providing clinically meaningful fracture assessments.

## 2.6 Evaluation Metrics

### 2.6.1 Concordance Index

The concordance index (C-index) is widely used to evaluate survival models and it assesses how accurately the model ranks each pair of subjects based on the actual times of events [27]. For each combination of pairs of subjects, the metric checks if the model predicts the events of the pair in the correct order, making the pair concordant, or incorrectly, making the pair discordant. Finally, the C-index is calculated as the proportion of concordant pairs out of all comparable pairs, meaning

that it ranges between 0 and 1, where a model with C-index greater than 0.5 has some predictive power and is better than guessing randomly [17].

However, this does not explain the issues of censoring, but the calculation of C-index handles that natively. This is done by simply discarding all pairs where both subjects are censored as there is no available event information. In the cases of pairs where one subject experienced an event and the other is censored, they are discarded if the time of censoring is less than the time of event, meaning that not enough information is available. However, if the time of event is less than the time of censoring, the pair can be evaluated and considered either concordant or discordant. The number of comparable pairs are then all pairs that were not discarded during this process, i.e. the sum of concordant and discordant pairs. The C-index can then be expressed:

$$\text{C-index} = \frac{\sum_{i,j} 1_{T_j<T_i} \cdot 1_{\eta_j>\eta_i} \cdot \delta_j}{\sum_{i,j} 1_{T_j<T_i} \cdot \delta_j}, \tag{2.23}$$

with

$$1_{T_j<T_i} = \begin{cases} 1 & \text{if } T_j < T_i \\ 0 & \text{otherwise} \end{cases}, \qquad 1_{\eta_j>\eta_i} = \begin{cases} 1 & \text{if } \eta_j > \eta_i \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

where $\eta_j$ is the risk score (model output), $T_j$ the event time (or censoring time) and $\delta_j$ the event indicator of the $j$th subject.

### 2.6.2 Area Under the Curve

The area under the curve (AUC) of the time-dependent receiver operating characteristic (ROC) will also be calculated as an evaluation method. The ROC represents the ability of the model to correctly distinguish between patients who experience a fracture at a certain time point, and those who do not [28]. In the ROC the true positive rate (TPR) will be plotted against the false positive rate (FPR) under different threshold values, which creates a curve. The AUC is then calculated, which represents the probability that if a patient with a fracture and a patient without a fracture are randomly chosen, the patient with the fracture will be given a higher risk score than the patient without the fracture. A higher AUC score, indicates a better performance of the model, if the AUC is larger than 0.5, it implies that the model is able to distinguish between positive and negative cases at a specific time. The time dependent AUC deals with censoring by considering it a non-event and omitting all pairs that consist of censoring and an event, no matter the order of the event and the censoring happening.

### 2.6.3 Brier Score

The Brier score (BS) makes use of the estimated probabilities of an event occurring rather than only the binary prediction like the C-index [29]. It can be expressed as:

$$\text{BS} = \frac{1}{N} \sum_{i=1}^{N} (f_i - \delta_i)^2 \tag{2.25}$$

where $N$ is the total number of predictions, $f_i$ the probability of a fracture for the $i$th subject, and $\delta_i$ is the event indicator for the $i$th subject (1 for a fracture having occurred, 0 otherwise). The definition means that the score ranges between 0 and 1, where 0 is the best score and implies no errors while 1 is the worst possible score. However, some adjustments are needed for the Brier score to be able to handle censoring. This can be done through inverse probability of censoring weighting (IPCW) [30]. Simply put, the score is scaled by a KM estimate of the survival function which upweights the subjects who are still observed at a time $t$. Equation 2.25 can then be modified

$$\mathrm{BS}(t) = \frac{1}{N}\sum_{i=1}^{N}\frac{(f_i(t) - \delta_i(t))^2}{\hat{S}(t_i)} \tag{2.26}$$

where $\hat{S}(t_i)$ is the KM estimate of the survival function for censoring at time $t_i$, which adjusts for censoring.

# 3

# Methods

## 3.1 Models

Several different models are developed in this project. Baseline models are constructed using only clinical risk factors represented as tabular data as input. These include a standard statistical model, and a deep machine learning implementation. Multimodal models using both images and tabular data as input are developed and implemented with different types of image input, architectures and fusion techniques.

### 3.1.1 Baselines

For baseline analysis, two models are considered. These are a Cox PH model and a DeepSurv model, both unimodal, taking only clinical risk factors as input. The Cox PH model is implemented using the `sksurv` library while DeepSurv is built with `PyTorch`, both in Python. Specifically, DeepSurv is designed with three fully connected layers, the first two followed by a ReLU activation function and dropout. The first layer takes an input size of 10, outputting 64 features. The second layer outputs 128 features while the final layer outputs 1 risk score for final prediction.

An alternative version of DeepSurv is used for some fusion models, described in upcoming sections. In this case, the final layer is omitted, such that the model instead outputs a size 128 feature vector used together with imaging features for final risk score prediction.

### 3.1.2 ConvDeepSurv

Convolutional neural networks trained for fracture risk prediction using the loss function defined in Section 2.2.5 are referred to as ConvDeepSurv (C-DS) models. All C-DS models developed in this project are based on the ResNet-50 architecture, implemented in `PyTorch` without pretrained ImageNet weights. Some architectural modifications were made to adapt the network to the data and task.

The original ResNet-50 expects 3-channel RGB input, but since the spinal X-ray images are in grayscale, the first convolutional layer was modified to accept a single-channel input. In another model variant that uses XVFA-extracted vertebral crops, 13 grayscale crops were stacked into a 13-channel input tensor, and the first convolutional layer was adjusted accordingly to accept 13 channels.

As the original classification layer in ResNet-50 outputs 1000 class logits, it was removed in all models. Instead, the 2048-dimensional feature vector from the penultimate layer was either used directly to predict a risk score by adding a single fully connected layer, or combined with outputs from other models (e.g., the tabular-only DeepSurv baseline) to enable multimodal learning.

### 3.1.3 ViTDeepSurv

Vision transformers trained for fracture risk prediction using the loss function defined in Section 2.2.5 are referred to as ViTDeepSurv (ViT-DS) models. ViT-DS was implemented using `PyTorch`, constructing the architecture from Dosovitskiy et. al. [11], with a few modifications. The model was only applied to XVFA-extracted vertebral crops and not full images. No pretrained weights were used.

Since the standard ViT expects 3-channel input images of size 224×224 pixels, the input projection layer was redefined to accept 13 channels, and the input image size was adjusted to 272×272 pixels to not lose too much resolution. This change required recomputing the number of patches and reinitializing the positional embedding to accommodate the resulting 64 tokens plus the class token.

Similarly to the ResNet-50 model, the standard ViT outputs 1000 class logits. This output is treated as a feature vector and is either used directly to predict a risk score by adding a single fully connected layer, or combined with outputs from other models to enable multimodal learning.

### 3.1.4 Intermediate Fusion Models

In order to make fracture risk predictions using multimodal data (i.e. images and tabular risk factors) a fusion based model is constructed, which will be referred to as IF-MLP. It fuses features from a image feature extractor backbone (either a C-DS or ViT-DS) and a separate fully connected network for the tabular data, commonly referred to as a multi-layer perceptron (MLP). The concatenated features are then passed through a final prediction head consisting of fully connected layers to produce a single risk score. Specifically, it concatenates the 2048 or 1000 image features (from C-DS and ViT-DS respectively) with the 128 tabular features into a single feature vector. This feature vector is passed through a fully connected layer with output size 128, before passing through batch normalization and a ReLU activation function. Finally, the feature vector is passed through another fully connected layer with output size 1 for final risk score prediction. An overview of how the model works with fusion and submodules is illustrated in Figure 3.1.

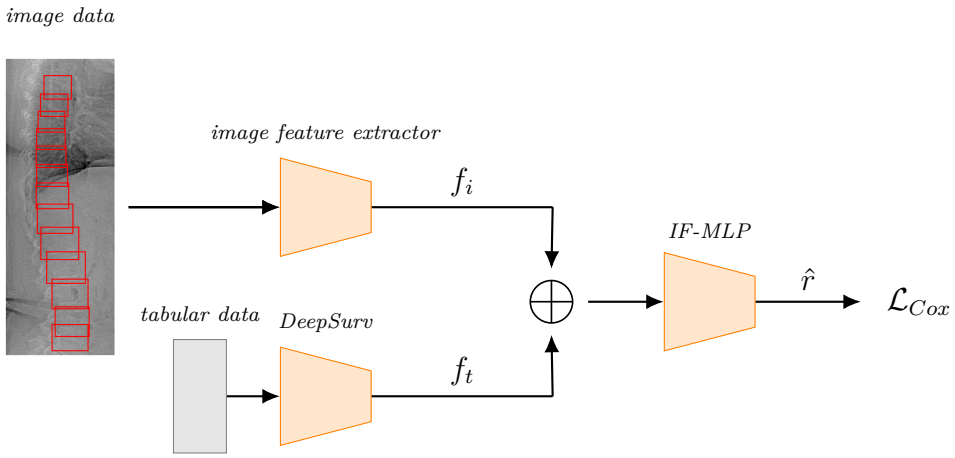



**Figure 3.1:** Intermediate fusion model overview. An image feature extractor (ConvDeepSurv or ViTDeepsurv) takes image data (full images or vertebral crops) as input. DeepSurv takes tabular data as input. Output features are concatenated and a final risk score prediction $\hat{r}$ is made by the intermediate fusion model.

### 3.1.5 Late Fusion Models

In the late fusion models, risk score predictions from an image model and the tabular DeepSurv model are used as input to predict a final risk score. Two methods of late fusion are considered.

The first approach, which will be referred to as the LF-FC model, is a fully connected layer that takes the two predicted risk scores as input and learns an optimal combination through training to output a final risk score. In this approach, only this fully connected layer is trained, i.e. the image model and tabular model are pretrained and frozen. An overview of this setup is illustrated in Figure 3.2.

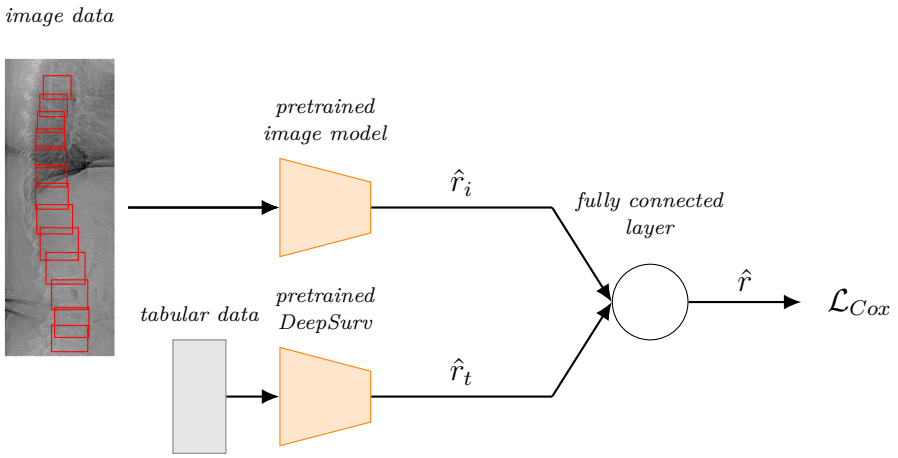



**Figure 3.2:** Late fusion with fully connected layer overview. A pretrained image model (ConvDeepSurv or ViTDeepsurv) takes image data (full images or vertebral crops) as input. Pretrained DeepSurv takes tabular data as input. Risk score outputs are used as input to the fully connected layer for final risk score prediction $\hat{r}$.

The other approach, which will be referred to as the $\alpha$-model, assumes a linear combination of the two predicted risk scores used as input, and combines them using a fixed parameter $\alpha$:

$$\hat{r} = \alpha \hat{r}_t + (1 - \alpha)\, \hat{r}_i, \tag{3.1}$$

where $\hat{r}$ is the predicted final risk score, $\hat{r}_t$ is the predicted risk score from the tabular model and $\hat{r}_i$ is the predicted risk score from the image model. This late fusion strategy provides a simple way to balance contributions from both data modalities. In this approach, the $\alpha$-model is trained for a set of $\alpha$-values, ranging from 0 to 1. The two submodules can then learn from each other and update themselves accordingly, which is not true for the first late fusion approach. An overview of how the $\alpha$-model works is illustrated in Figure 3.3.

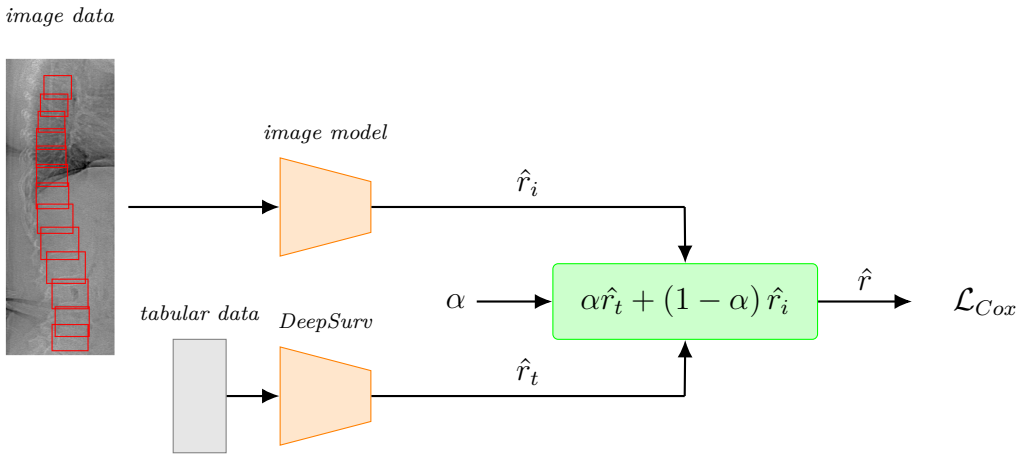



**Figure 3.3:** $\alpha$-model overview. An image model (ConvDeepSurv or ViTDeepsurv) takes image data (full images or vertebral crops) as input and predicts a risk score. DeepSurv predicts a risk score based on tabular input. The final risk score $\hat{r}$ is computed by combining the two predicted risk scores through their relationship with the parameter $\alpha$.

### 3.1.6 Explainable Vertebral Fracture Analysis

The vertebra detection module from XVFA [26] is utilized to extract coordinates of bounding boxes in the spinal X-ray images. These bounding boxes depict crops of individual vertebrae, which are then used as input to the models. The detection module locates a variable amount of vertebrae depending on patient, whereas a fixed number is needed in order to keep input size to the models constant. This is handled by filtering - discarding additional boxes or adding empty boxes for each patient until the desired number of crops is met.

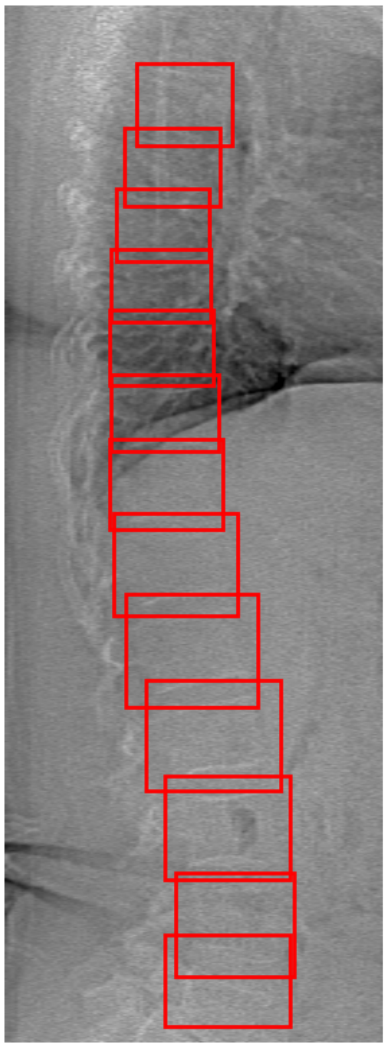


**Figure 3.4:** Example of predicted bounding boxes from XVFA.

Figure 3.4 depicts a spinal X-ray of a patient with 13 detected bounding boxes from XVFA. As can be seen, boxes are of varying shape, simply dependent on the sizes of individual vertebrae.

## 3.2 Evaluation

Models are evaluated in terms of predictive performance compared to baselines as well as to other types of developed models. The evaluation aims to determine how well each model can predict risk scores for fractures, using metrics that capture different aspects of model quality.

To measure the models' performance objectively, the dataset is split into a training set and a test set. The test set is held out during training and therefore unseen to the models at the time of evaluation. Cross-validation is used during training for the training set. With cross-validation, the training set is split into $k$ folds of equal size. The model is trained $k$ times, where each fold of data forms a validation set once,

while the other $k-1$ folds form a training set. This means that each model type obtain $k$ fold-specific models, each validated on different data. In order to evaluate the model type in cross-validation, one can combine the results of each fold-specific model, e.g. by means of average across all folds with standard deviation reflecting uncertainty. The use of cross-validation can allow for better model generalization than the standard training-validation-test setup since it is not as dependent on the initial data split.

As part of the evaluation, performance is compared between models using only tabular risk factors and those incorporating both tabular and image data, to assess whether multimodal inputs improve prediction accuracy. Additionally, models using full spinal images are compared with those using localized vertebral crops extracted by the XVFA module.

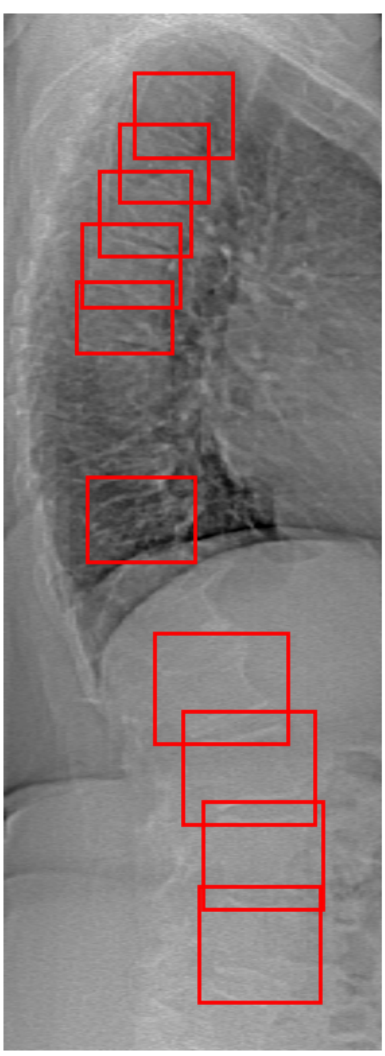
## 3.3 Experiments

Cross-validation is used during training and evaluation. Specifically, 10% of the dataset was held out for final evaluation, while the remaining data was used in a 3-fold cross-validation setup, where models are trained on different subsets and evaluated based on an average performance across folds.

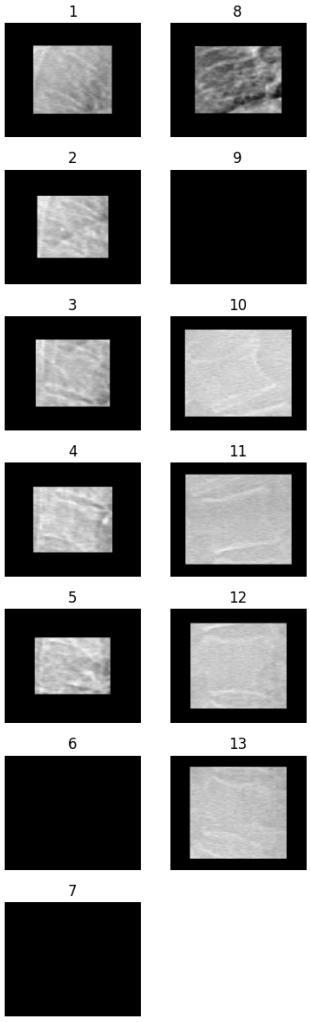
Before training, preprocessing is applied to standardize input data. For the tabular data, the risk factors age and BMI were scaled using a standard scaler by removing the mean and scaling to unit variance, the scaler fitted to the training set for each fold.

**Preprocessing of full images**
As the images are of varying sizes and rather large, they are normalized by the maximum and minimum pixel value across the current fold and then resized to a fixed resolution of 1024×512 pixels. In order to keep the image size constant while allowing rotation of $\pm 15^{\circ}$ in the augmentation, all images are padded to 1121×759 pixels. This is simply the smallest size that allow for all possible rotations in the specified range without losing information to cropping.

**Preprocessing of cropped images**
For each patient, 13 crops of vertebrae are used. However, the vertebra detection module from XVFA does not always produce 13 boxes, nor do they always contain only one vertebra in each bounding box. Through visual examination, bounding boxes with height greater than 230 pixels were discarded since they were incorrect predictions, containing multiple vertebrae. In the cases where more than 13 were detected, the highest box (in the patient's neck) was discarded until 13 boxes were left. For the patients with less than 13 vertebrae detected, empty boxes were added in the positions were the distance between detected boxes exceeded a threshold, see Figure 3.5. All crops were padded to 230×272 pixels, based on the largest width and height in the dataset (after the previous filtering), and stacked into an image containing the 13 cropped vertebrae in a channel each.



**(a)** Example of too few predicted bounding boxes from XVFA.




**(b)** Example of crops used along with added empty crops in predicted positions, numbered from top to bottom.

**Figure 3.5:** Example of a bad prediction from XVFA with correction used for input.

Augmentation is applied during training, including random dropout (salt and pepper noise), rotation ($\pm 15°$) and vertical flipping for the full images. Rotation and flipping is helpful since the models learn relationships between features in relation to other features, rather than relationships between features and positions. For the cropped images, augmentation included vertical and horizontal flipping, zooming, Gaussian noise and contrast adjustment. The augmentation helps in avoiding overfitting, and is necessary for model generalization since the dataset is not very large.

The baseline models are a standard CPH model and a DeepSurv model. They are implemented in Python, using the `scikit-survival` library for the standard CPH model and `PyTorch` for the DeepSurv model. All other models, ConvDeepSurv, ViTDeepSurv, intermediate fusion models and late fusion models are implemented using `PyTorch`. Mainly, ConvDeepSurv uses `resnet50` with no pretrained weights

as base, while ViTDeepsurv uses `vit_b_32` with no pretrained weights.

The models were trained on GPUs, and the `PyTorch` library was used to construct datasets, data loaders and optimizers. The loss function used for updating the models is the Cox partial likelihood, defined in Section 2.2.5. For each model architecture, a model was trained on each separate fold of training data, with corresponding validation set. The C-index was calculated on the validation set every two epochs, and both the final iteration of the model and the best performing in terms of validation C-index are saved. The latter is then used for final evaluation.

A few of the models had additional specific settings for training. The IF-MLP models are trained end-to-end, such that the gradient is flowing through all submodules. For the late fusion models, the LF-FC model only trained its final fully connected layer, while the models for extracting the image risk score and tabular risk score were pretrained and frozen. As for the other approach with the $\alpha$-model, a sweep was made over a set of values for $\alpha$ (0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9). For each $\alpha$, the model is fully trained on each fold and the $\alpha$-model chosen to evaluate is the one with best performing mean validation C-index across all three folds. Further, the $\alpha$-model is utilized in two different ways. One with end-to-end training, and another with the tabular model being pretrained and frozen, such that only the image model's parameters are updated. All hyperparameters used for training of each model can be found in Appendix, Table A.1.

# 4

# Results & Discussion

## 4.1 Baseline

The DeepSurv baseline implementation for the tabular data proved to be insensitive to hyperparameters during training, and was able to display moderate performance on both validation and test sets. The evaluation metrics are shown along with the Cox model baseline results in Table 4.1 and 4.2.

**Table 4.1:** Baseline performance on the validation set.

| Model | C-index | AUC | BS |
|---|---|---|---|
| *DeepSurv* | 0.59 ± 0.00 | 0.62 ± 0.00 | 0.16 ± 0.00 |
| *Cox PH* | 0.61 ± 0.01 | 0.63 ± 0.02 | 0.20 ± 0.01 |

**Table 4.2:** Baseline performance on the test set.

| Model | C-index | AUC | BS |
|---|---|---|---|
| *DeepSurv* | 0.59 ± 0.01 | 0.64 ± 0.01 | 0.17 ± 0.00 |
| *Cox PH* | 0.63 ± 0.00 | 0.66 ± 0.01 | 0.20 ± 0.01 |

The Cox model achieves higher C-index and AUC values on both the validation and test sets, indicating stronger discrimination in ranking individuals by fracture risk. In contrast, DeepSurv attains lower Brier scores, suggesting better calibration, which means its predicted probabilities more closely reflect the true event rates.

## 4.2 Ablation study

To evaluate the effectiveness of combining spinal X-rays and clinical risk factors for fracture prediction, multiple fusion strategies, input image formats, and model architectures were tested. These experiments are summarized, along with baselines in Table 4.3. The baselines include traditional CPH and DeepSurv models using only tabular data. The C-DS models were tested with different inputs: full spinal X-ray images, vertebral crops extracted using the XVFA method, and multimodal combinations of these images with clinical risk factors using both intermediate and

late fusion strategies. Additionally, ViT-DS models were evaluated using the XVFA-extracted crops, using both intermediate and late fusion strategies.

**Table 4.3:** Model performance on test data. C-DS indicates ConvDeepSurv architecture, ViT-DS indicates Vision transformer Deepsurv architecture. Bold indicates the highest performance achieved in each category. (*) indicates pretrained modality prediction module.

| Model | Crops | Fusion | | | Evaluation Metric | | |
|---|---|---|---|---|---|---|---|
| | | Interm. | Late | | | | |
| | | MLP | MLP | $\alpha$ | C-index | AUC | BS |
| *C-DS (end-to-end)* | | ✓ | | | 0.57 ± 0.03 | 0.61 ± 0.03 | 0.17 ± 0.01 |
| | ✓ | ✓ | | | **0.69 ± 0.04** | **0.76 ± 0.05** | **0.14 ± 0.01** |
| | ✓ | | | ✓ | 0.64 ± 0.02 | 0.68 ± 0.04 | 0.16 ± 0.01 |
| *C-DS (img.*, tab.*)* | | | ✓ | | **0.67 ± 0.03** | **0.73 ± 0.04** | 0.19 ± 0.05 |
| *C-DS (img.*, tab.*)* | ✓ | | ✓ | | 0.66 ± 0.03 | 0.71 ± 0.04 | 0.17 ± 0.02 |
| *C-DS (tab.*)* | ✓ | | | ✓ | 0.61 ± 0.02 | 0.65 ± 0.02 | **0.16 ± 0.00** |
| *ViT-DS (end-to-end)* | ✓ | ✓ | | | 0.64 ± 0.04 | 0.69 ± 0.05 | **0.15 ± 0.01** |
| | ✓ | | | ✓ | **0.66 ± 0.01** | **0.70 ± 0.01** | **0.15 ± 0.01** |
| *C-DS* | | *images only* | | | 0.59 ± 0.06 | 0.63 ± 0.08 | 0.20 ± 0.03 |
| | ✓ | | | | **0.65 ± 0.04** | **0.69 ± 0.04** | **0.17 ± 0.02** |
| *DeepSurv* | | *tabular only* | | | 0.59 ± 0.01 | 0.64 ± 0.01 | **0.17 ± 0.00** |
| *Cox PH* | | | | | **0.63 ± 0.00** | **0.66 ± 0.01** | 0.20 ± 0.01 |

Further experimentation with certain models and setups was guided by validation performance, presented in Appendix, Table A.2. For the $\alpha$-models, $\alpha = 0.8$ yielded the best performance on the validation data for all cases where ConvDeepSurv (C-DS) was the image prediction module, while $\alpha = 0.9$ was optimal for ViTDeepSurv (ViT-DS). Based on the validation results, the $\alpha$ values yielding the best performance for each image prediction module were chosen for final evaluation on the test set.

Overall, the results demonstrate that images contain relevant information for fracture risk prediction on their own. Models using only images for prediction achieved performance comparable to baseline models, indicating that radiographic features alone can be strong predictors of fracture outcomes. Multimodal models generally outperformed baselines across all evaluation metrics, supporting the hypothesis that incorporating images provides complementary information that enhances prediction. Both intermediate and late fusion approaches yielded competitive results, suggesting that multiple fusion strategies are valid and effective.

The best performing model overall was the end-to-end trained C-DS crop model with intermediate fusion. It achieved a C-index of 0.69, AUC of 0.76, and Brier score of 0.14. Compared to the best baseline (Cox PH, C-index 0.63, AUC 0.66, DeepSurv, BS 0.17), this represents a substantial improvement across all metrics. Not only does it display the effect of multimodal learning, it also demonstrate the value of jointly training the submodules.

It is also worth noting that the best performing models on validation data did not

always generalize as well to the test set. Evidently, $\alpha$-models, using late fusion, display the best performance on validation data. That is not the case for the test data, which might indicate slight overfitting.

## 4.3 Image Input Strategy

The impact of image input strategy is of great interest. Across the image only C-DS models, it is clear that using XVFA-extracted crops is preferable to full spinal X-ray images. As expected, the model using crops as input performs better in all evaluation metrics, as the image-only C-DS model improved from a C-index of 0.59 with full images to 0.65 with crops. This indicates that the cropped images provide stronger signal, and support the hypothesis that reducing irrelevant background and focusing on local vertebral features enhances predictive performance.

Similarly for the C-DS models trained end-to-end, crops as input outperform using full images as input. Here, the full-image model performed worse than baselines. One possible reason, in this setup, could be that the full images dilute relevant features and introduce more background noise. In turn, this could cause the image features to dominate prediction due to stronger gradients, causing the tabular features to be less utilized. Usage of crops could solve such issues by forcing the image feature extractor to focus on vertebral features.

Inversely, when using late fusion with pretrained and frozen submodules (using the LF-FC model), the C-DS model trained on full spinal images performed exceptionally well. This may be because training only a single fully connected layer rather than updating the entire model reduces gradient noise, leading to more stable and effective optimization. This result suggests that late fusion with full images can be highly effective when the submodules are sufficiently well-trained.

## 4.4 Model Architecture Comparison

Comparing image model architectures, the best C-DS models outperforms the best ViT-DS model across all comparable configurations. A reason for this could be that convolutional filters, which focuses on local spatial patterns, are particularly effective for detecting features important for fracture risk prediction. Even though less experiments and tuning was done with ViT-DS, they still perform competitively. With further optimization, it is likely that ViT-DS models could perform on par with the best C-DS models.

# 5

# Conclusion

This project has proven that the incorporation of multimodal learning can improve fracture risk prediction. The results demonstrate that image data alone contain meaningful fracture risk signal. However, by integrating image data with clinical risk factors in multimodal models, both the DeepSurv and Cox PH baselines were consistently outperformed. Furthermore, the findings suggest that using vertebral crops is generally more effective than using full spinal X-ray images. The comparative analysis of neural network architectures yielded encouraging results, though the question of whether CNNs or ViTs are preferable remains partially unsolved. This is primarily due to the limited experiments with ViTs tuning and design in this study. Nonetheless, both architectures display promising results.

Among all models, the best performance was achieved by the end-to-end trained ConvDeepSurv model using vertebral crops and intermediate MLP fusion, outperforming the strongest baseline in all evaluated metrics, achieving a C-index of 0.69 versus 0.63, an AUC of 0.76 versus 0.66, and a Brier score of 0.14 versus 0.17. These results support the effectiveness of combining image features and clinical risk factors in a joint learning framework.

Although the results are promising, they are based on models trained and evaluated solely on a dataset of older Caucasian women, which limits generalizability. While difficult to obtain, future work could focus on improving generalizability by using larger datasets, incorporating more diverse populations, though such datasets are currently not available. Methods of incorporating explainability and interpretability could be explored, as these aspects are important for employing models in clinical practice. Future work should continue to investigate CNN-based approaches, which performed strongly in this study. Additionally, further exploration of ViTs is warranted, as time constraints limited the number of experiments conducted with them in this project.

# Bibliography


[1] M. Lorentzon, A. G. Nilsson, H. Johansson, J. A. Kanis, D. Mellström, and D. Sundh, "Extensive undertreatment of osteoporosis in older Swedish women," *Osteoporosis International*, vol. 30, no. 6, pp. 1297–1305, 2019, ISSN: 14332965. DOI: 10.1007/s00198-019-04872-4.

[2] T. Sozen, L. Ozisik, and N. Calik Basaran, "An overview and management of osteoporosis," *European Journal of Rheumatology*, vol. 4, no. 1, pp. 46–56, 2017, ISSN: 21479720. DOI: 10.5152/eurjrheum.2016.048.

[3] R. Bernabei, A. M. Martone, E. Ortolani, F. Landi, and E. Marzetti, "Screening, diagnosis and treatment of osteoporosis: A brief review," *Clinical Cases in Mineral and Bone Metabolism*, vol. 11, no. 3, pp. 201–207, 2014, ISSN: 19713266. DOI: 10.11138/ccmbm/2014.11.3.201.

[4] J. A. Kanis, O. Johnell, A. Oden, B. Jonsson, A. Dawson, and W. Dere, "Risk of hip fracture derived from relative risks: An analysis applied to the population of Sweden," *Osteoporosis International*, vol. 11, no. 2, pp. 120–127, 2000, ISSN: 0937941X. DOI: 10.1007/PL00004173.

[5] J. A. Kanis, O. Johnell, A. Oden, H. Johansson, and E. McCloskey, "FRAX™ and the assessment of fracture probability in men and women from the UK," *Osteoporosis International*, vol. 19, no. 4, pp. 385–397, 2008, ISSN: 0937941X. DOI: 10.1007/s00198-007-0543-5.

[6] A. S. Areeckal, M. Kocher, and S. David, "Current and Emerging Diagnostic Imaging-Based Techniques for Assessment of Osteoporosis and Fracture Risk," *IEEE Reviews in Biomedical Engineering*, vol. 12, pp. 254–268, 2018, ISSN: 19411189. DOI: 10.1109/RBME.2018.2852620.

[7] T. M. Link, "Osteoporosis imaging: State of the art and advanced imaging," *Radiology*, vol. 263, no. 1, pp. 3–17, 2012, ISSN: 00338419. DOI: 10.1148/radiol.12110462.

[8] Y. C. Lee, J. Cha, I. Shim, *et al.*, "Multimodal deep learning of fundus abnormalities and traditional risk factors for cardiovascular risk prediction," *npj Digital Medicine*, vol. 6, no. 1, 2023, ISSN: 23986352. DOI: 10.1038/s41746-023-00748-4.

[9] N. Lessmann, B. van Ginneken, P. A. de Jong, and I. Išgum, "Iterative fully convolutional neural networks for automatic vertebra segmentation and identification," *Medical Image Analysis*, vol. 53, pp. 142–155, 2019, ISSN: 13618423. DOI: 10.1016/j.media.2019.02.005. [Online]. Available: https://doi.org/10.1016/j.media.2019.02.005.

[10] T. Y. Yen, C. S. Ho, Y. C. Pei, *et al.*, "Predicting osteoporosis from kidney-ureter-bladder radiographs utilizing deep convolutional neural networks," *Bone*, vol. 184, no. 5, p. 117 107, 2024, ISSN: 87563282. DOI: 10.1016/j.bone.2024.117107. [Online]. Available: https://doi.org/10.1016/j.bone.2024.117107.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "an Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale," *ICLR 2021 - 9th International Conference on Learning Representations*, 2021.

[12] J. Chen, J. Mei, X. Li, *et al.*, "TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers," *Medical Image Analysis*, vol. 97, no. February, 2024, ISSN: 13618423. DOI: 10.1016/j.media.2024.103280.

[13] A. Shaban and S. Yousefi, "Multimodal Deep Learning," *Springer Optimization and Its Applications*, vol. 211, pp. 209–219, 2024, ISSN: 19316836. DOI: 10.1007/978-3-031-53092-0{_}10.

[14] S. C. Huang, A. Pareek, S. Seyyedi, I. Banerjee, and M. P. Lungren, "Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines," *npj Digital Medicine*, vol. 3, no. 1, 2020, ISSN: 23986352. DOI: 10.1038/s41746-020-00341-z. [Online]. Available: http://dx.doi.org/10.1038/s41746-020-00341-z.

[15] S. Qiu, M. I. Miller, P. S. Joshi, *et al.*, "Multimodal deep learning for Alzheimer's disease dementia assessment," *Nature Communications*, vol. 13, no. 1, pp. 1–17, 2022, ISSN: 20411723. DOI: 10.1038/s41467-022-31037-5.

[16] H. K. Genant, Y. W. Chun, C. van Kuijk, and M. C. Nevitt, "Vertebral Fracture Assessment Using a Semiquantitative Technique," *Journal of Bone and Mineral Research*, vol. 8, no. 9, pp. 1137–1148, 1993.

[17] D. G. Kleinbaum and M. Klein, *Survival analysis: A self-learning text*, Third edit. Springer Science+Business Media, LLC, 2012, ISBN: 9780387244471. DOI: 10.1007/978-0-387-49979-6{_}25.

[18] M. Goel, P. Khanna, and J. Kishore, "Understanding survival analysis: Kaplan-Meier estimate," *International Journal of Ayurveda Research*, vol. 1, no. 4, p. 274, 2010, ISSN: 0974-7788. DOI: 10.4103/0974-7788.76794.

[19] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network," *BMC Medical Research Methodology*, vol. 18, no. 1, pp. 1–12, 2018, ISSN: 14712288. DOI: 10.1186/s12874-018-0482-1.

[20] Y. c. Wu and J. w. Feng, "Development and Application of Artificial Neural Network," *Wireless Personal Communications*, vol. 102, no. 2, pp. 1645–1656, 2018, ISSN: 1572834X. DOI: 10.1007/s11277-017-5224-x. [Online]. Available: https://doi.org/10.1007/s11277-017-5224-x.

[21] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. DOI: 10.1162/neco.1989.1.4.541.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "LeNet," *Proceedings of the IEEE*, no. November, pp. 1–46, 1998, ISSN: 00189219.

[23] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017, ISSN: 10495258.

[24] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[25] X. Zhu, J. Yao, and J. Huang, "Deep convolutional neural network for survival analysis with pathological images," *Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016*, no. 2, pp. 544–547, 2017. DOI: 10.1109/BIBM.2016.7822579.

[26] V. Wåhlstrand Skärström, J. Alvén, L. Johansson, M. Lorentzon, and I. Häggström, "Explainable vertebral fracture analysis with uncertainty estimation using differentiable rule-based classification," *Miccai 2024*, vol. in submiss, pp. 1–13,

[27] A. R. Brentnall and J. Cuzick, "Use of the concordance index for predictors of censored survival data," *Statistical Methods in Medical Research*, vol. 27, no. 8, pp. 2359–2373, 2018, ISSN: 14770334. DOI: 10.1177/0962280216680245.

[28] C. Marzban, "The ROC curve and the area under it as performance measures," *Weather and Forecasting*, vol. 19, no. 6, pp. 1106–1114, 2004, ISSN: 08828156. DOI: 10.1175/825.1.

[29] M. S. Roulston, "Performance targets and the Brier score," *Meteorological Applications*, vol. 14, no. 2, pp. 185–194, 2007, ISSN: 14698080. DOI: 10.1002/met.21.

[30] H. Kvamme and Ø. Borgan, "The Brier Score under Administrative Censoring: Problems and Solutions," vol. 24, pp. 1–26, 2019, ISSN: 15337928. [Online]. Available: http://arxiv.org/abs/1912.08581.

# A

# Appendix

**Table A.1:** Hyperparameters used during training. C-DS indicates ConvDeepSurv architecture, ViT-DS indicates Vision transformer Deepsurv architecture. (*) indicates pretrained modality prediction module. In all cases, Adam was used as optimizer. In the cases of scheduler being used, Cosine annealing was used with `T_max=n Epochs` and `eta_min=0.1·lr`.

| Model | Crops | Fusion | | | Hyperparameter | | | |
|---|---|---|---|---|---|---|---|---|
| | | Interm. | Late | | | | | |
| | | MLP | MLP | $\alpha$ | lr | Scheduler | n Epochs | Batch size |
| *C-DS (end-to-end)* | | ✓ | | | 1e-4 | ✓ | 200 | 32 |
| | ✓ | ✓ | | | 7e-5 | | 200 | 128 |
| | ✓ | | | ✓ | 1e-4 | ✓ | 100 | 128 |
| *C-DS (img.*, tab.*)* | | | ✓ | | 2e-2 | | 20 | 32 |
| *C-DS (img.*, tab.*)* | ✓ | | ✓ | | 2e-2 | | 20 | 128 |
| *C-DS (tab.*)* | ✓ | | | ✓ | 1e-4 | ✓ | 50 | 128 |
| *ViT-DS (end-to-end)* | ✓ | ✓ | | | 1e-5 | ✓ | 150 | 64 |
| | ✓ | | | ✓ | 1e-5 | ✓ | 50 | 64 |
| *C-DS* | | *images only* | | | 1e-4 | ✓ | 150 | 32 |
| | ✓ | | | | 1e-4 | ✓ | 200 | 128 |
| *DeepSurv* | | *tabular only* | | | 1e-3 | | 10 | 32 |

**Table A.2:** Model performance on validation data. C-DS indicates ConvDeepSurv architecture, ViT-DS indicates Vision transformer Deepsurv architecture. Bold indicates the highest performance achieved in each category. (*) indicates pretrained modality prediction module.

| Model | Crops | Fusion | | | Evaluation Metric | | |
|---|---|---|---|---|---|---|---|
| | | Interm. | Late | | | | |
| | | MLP | MLP | $\alpha$ | C-index | AUC | BS |
| *C-DS (end-to-end)* | | ✓ | | | 0.60 ± 0.02 | 0.63 ± 0.02 | **0.17 ± 0.00** |
| | ✓ | ✓ | | | 0.58 ± 0.01 | 0.61 ± 0.02 | **0.17 ± 0.00** |
| | ✓ | | | ✓ | **0.61 ± 0.01** | **0.63 ± 0.01** | 0.17 ± 0.02 |
| *C-DS (img.*, tab.*)* | | | ✓ | | 0.58 ± 0.03 | 0.61 ± 0.03 | 0.21 ± 0.03 |
| *C-DS (img.*, tab.*)* | ✓ | | ✓ | | 0.58 ± 0.01 | 0.60 ± 0.01 | 0.20 ± 0.05 |
| *C-DS (tab.*)* | ✓ | | | ✓ | **0.60 ± 0.01** | **0.63 ± 0.01** | **0.16 ± 0.00** |
| *ViT-DS (end-to-end)* | ✓ | ✓ | | | **0.57 ± 0.01** | **0.59 ± 0.02** | **0.17 ± 0.00** |
| | ✓ | | | ✓ | 0.56 ± 0.02 | 0.57 ± 0.03 | 0.18 ± 0.02 |
| *C-DS* | | *images only* | | | 0.55 ± 0.02 | **0.59 ± 0.03** | 0.20 ± 0.03 |
| | ✓ | | | | **0.56 ± 0.01** | 0.58 ± 0.02 | **0.19 ± 0.05** |
| *DeepSurv* | | *tabular only* | | | 0.59 ± 0.00 | 0.62 ± 0.00 | **0.16 ± 0.00** |
| *Cox PH* | | | | | **0.61 ± 0.01** | **0.63 ± 0.02** | 0.20 ± 0.01 |

**DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden
www.chalmers.se


CHALMERS
UNIVERSITY OF TECHNOLOGY