VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS STUDY PROGRAM

MASTER THESIS

Survival Analysis Incorporating Medical Imaging Data

Medicininių vaizdų panaudojimas išgyvenamumo analizėje

Ieva Jonaitytė

Supervisor Assist. Dr. Linas Petkevičius

Vilnius 2021

Medicininių vaizdų panaudojimas išgyvenamumo analizėje

Santrauka

Šiame darbe nagrinėjama medicininių nuotraukų panaudojamumas išgyvenamumo analizėje. Vis dar įprasta nuotraukų analizėje tyrinėti požymius, kuriuos apibrėžia tyrėjas. Pastarajam atvejui yra reikalingos srities žinios, taip pat skirtingoms užduotims požymių parinkimo logika gali stipriai skirtis, yra individuali ir subjektyvi. Tyrime analizuojamas giliųjų požymių poveikis išgyvenamumo analizei. Pirmiausia nagrinėjami modeliai, grįsti konvoliucinių neuronų tinklų struktūromis, išskiriami galimai reprezentatyvūs požymiai - nauji aiškinamieji kintamieji. Toliau iš jų suformuojami duomenų rinkiniai Kokso proporcingų rizikų regresijos analizei. Tikslas - ištirti, ar pasiūlyti modeliai geba suformuoti informatyvius požymius, kurie būtų reikšmingi išgyvenamumo modeliui, nenaudojant išankstinio požymių apsibrėžimo ar papildomo vaizdų žymėjimo. Darbe sukurti konvoliucinių neuroninių tinklų modeliai, generuojantys informatyvius požymius iš vaizdų. Atlikti eksperimentiniai tyrimai naudojant krūties vėžio duomenis.

Raktiniai žodžiai : Medicininių vaizdų analizė, histopatologinės nuotraukos, konvoliuciniai naurnų tinklai, išgyvenamumo analizė, Kokso proporcingų rizikų regresija, cenzūravimas

Survival Analysis Incorporating Medical Imaging Data

Abstract

In this thesis the possibility of using medical images for survival analysis is investigated. In such analysis it is still common to use handcrafted features which needs lots of prior knowledge. Also such processing is quite different and subjective for each specific task. In this research we analyse how deep features affect survival model. Firstly, a few methods using different convolutional models are applied to obtain the representative covariates. Second, these image-related covariates are analysed by Cox proportional hazard regression. The aim is to find out whether proposed models are capable of extracting any meaningful covariates that are significant in survival analysis, without using explicit feature engineering or image labelling. During the work convolutional neural network models generating deep features from images are created. The experiments are run on breast cancer data set.

Key words : Medical image analysis, Histopathology images, Convolutional Neural Networks, Survival analysis, Cox proportional hazard regression, censoring

Table of Contents

Introduction

In recent years with the latest deep learning (DL) technologies there has been a breakthrough in computer vision. The convolutional neural network success in ImageNet challenge is often referred to as a great starting point [2]. Since computer vision has greatly evolved there is a natural trend to integrate image analysis into any field that is capable of producing images so the applications spread in cancer diagnostics as well [38]. The specifics of these algorithms empower feature extraction without the necessity of explicit feature engineering. The design of handcrafted features is often individualized and often quite different for each task. Deep learning algorithms ability to transform input data into a more abstract and composite representations result in better performance and makes the their usage less dependent on the prior knowledge and feature engineering skill [3]. However these methods show less interpretable feature extraction and intermediate results, which play a fundamental role in research and application [30].

Large image databases of many health topics are available in the digital age: for example NIH Database of 100,000 Chest X-Rays [34], Alzheimer's Disease Neuroimaging Initiative [14], OASIS brain data sets[21] and other. Most popular modalities of medical images are: radiography, magnetic resonance, computed tomography, ultrasonic and nuclear images. Most common problems solved using diagnostic medical imaging is tissue/object classification, anomaly detection, quality correction [15]. Methods for these problem solution begin with the mentioned feature design - the search of meaningful characteristics however since the usage of convolutional neural networks showed significantly higher capabilities of feature extraction from images, the effort to apply them in medical imaging started as well. These methods are still conditionally new to the field, first scientific papers showing meaningful results and broad application possibilities start from around 2016. The first trials to use medical images for survival analysis start even later at 2018 [22]. The possibility to investigate survival from image data in general is strongly narrowed by the demand of both aspects: images and time-to-event information. So far such analyses cover cancer, Alzheimer disease [24].

Powerful automatic image analysis is relevant to medical field:

- Microscopic image analysis can be time consuming for a human. Therefore, the methodology or tools that could reduce the time is of great relevance.

- Computer vision can sometimes even surpass a human eye, so might lead to new findings. As an example, face recognition. Although we might think that we can distinguish different persons and recognize same efficiently (because we use lots of meaningful patterns), a study shows that the state-of-the-art algorithms perform better than average recognizer and equally as so called super-recognizers (persons trained for facial recognition) [23].

- Deep learning models can achieve very good results however a great benefit for the specialists and research would be an effective methodology for intermediate result visualization and interpretation.

Diagnostic medicine is one of the fields where it is difficult to have enough labeled data, because most of the time labeling has to be done by a specialist. Therefore it is beneficial to seek for methods using most simple/natural annotations, meaning that the algorithms should run on data of some form that is easily obtainable by specialists, institutions. For example, it is quite expensive to demand large databases of segmented medical pictures (where the contours are marked).

The aim of the thesis

The aim of the thesis: investigate how meaningful information can be added to survival model from raw tissues images without full size image analysis, data labeling and having limited computing resources.

Main objectives are formulated as follows:

- Investigate the scientific literature of related works
- Collect and prepare the cancer tissue imaging data
- Construct medical images compression models
- Investigate impact of encoded covariates for Cox proportional hazards regression and provide recommendations of feature extraction.

1 Related Work

Scientific paper selection for literature analysis was selected by topics: diagnosing cancer from images, survival analysis from images of any field, working with large images. The main criteria was latest publications, journal popularity followed by further analysis of mentioned methods.

1.1 Computer vision

Computer vision covers automatic methods for image or other visual material recognition. This field tries to solve problems of highly complex nature and is very closely related to machine learning.

In recent years deep learning techniques have become most popular computer vision methods. However there are other groups of algorithms used in image processing, such as point operations, linear filtering, neighborhood operations, Fourier, geometric transformations and other depending on a problem [29]. The goal in general is to extract meaningful features from visual material and to convert them into structured information.

Here are a few examples of formulated computer vision tasks applied in recognition of photo content:

- Object Classification - what object is in the image [16]
- Object Verification - is some object in the image [27]
- Object Detection - what is the location of some object in the image [8]
- Object Landmark Detection: what are the representative points of some object and where they are located in the image [35]
- Object Segmentation: locating all the pixels belonging to some object in the image [20]

1.2 Medical image analysis

The algorithms depending on the task specifics may vary quite broadly. In terms of visual feature research applied in medical image classification we may name 3 levels: low-level feature extraction, mid-level feature representation, and deep feature learning. The low-level feature extraction methods mainly describe image content in the aspect of color, shape, texture. In this stage color vector patterns [10], SIFT, simple linear iterative clustering (SLIC)[4] and local binary patterns (LBP) [11] are used. The mid-level methods process and encode low-level feature extraction results and may carry semantic information. For example bag-of-visual-words (BoVW)[37]. Conditionally recent deep learning techniques may extract next level features (deep features) from original image pixels without any initial assumptions [19].

Not so long even last 5 years ago low and mid-level feature construction in medical image analysis was common approach. Moreover, analysis using these so called handcrafted features based on the knowledge of the problems nature is still quite popular. In the photos of cells some color and shape patterns are rather common, for example nuclei are often of more saturated darker color. Using this pre-knowledge, a study mentions a blue ratio thresholding used for selecting candidate regions for mitosis detection [33].

It is possible to use a combination of machine learning and the handcrafted features, but recent breakthroughs in deep learning indicate that high-level features learned from data works better in practice.

1.3 Extremely large image problem

First aspect to be considered is the applied ways of managing extremely large images which are the case of cancer imaging. Having limited computing resources it is not effective or even possible to extract features with Convolution Neural Networks directly from this size of image. Commonly CNN models due to technical properties and data specifics work with fixed size mid-size images eg. 512x512. For that reason when medical images are larger various approaches are used.

1.3.1 Image patching

Reducing the size of the image means that discriminative details are lost therefore one popular approach is image patching. The large image is divided into a grid and a sliding window approach is applied to extract all the patches for the CNN training [6].

1.3.2 Different approaches to image patch analysis

For many tasks (such as segmentation of tissue primitives, anomaly detection and other) effective patch sampling is necessary. Deep learning model effectiveness depends on provided data, so it is important to select representative patches of the image for training. Patch sampling techniques may reach from random [36] to segmentation-guided sampling.

Another approach possible is to analyze each patch, this way keeping the low loss of data. For example one possible way is cellular analysis, where the descriptors are composed using hand-crafted image features. [7].

1.4 Image patching using deep learning models

A recent work [36] proposes a direct way to train a deep learning survival model from images.

Here a random 16 patches of 256x256 pixels are sampled and analyzed as a single batch. The authors claim, that supposing density of tumor cells is large enough, a number of randomly selected patches cover informative areas by quite high probability and represent the tissue sufficiently.

The presented model consists of multiple convolutional neural network (CNN) modules with shared weights, and an average pooling layer that merges image features computed by these modules. It is supposed to filter only the informative patch information see Figure 1.

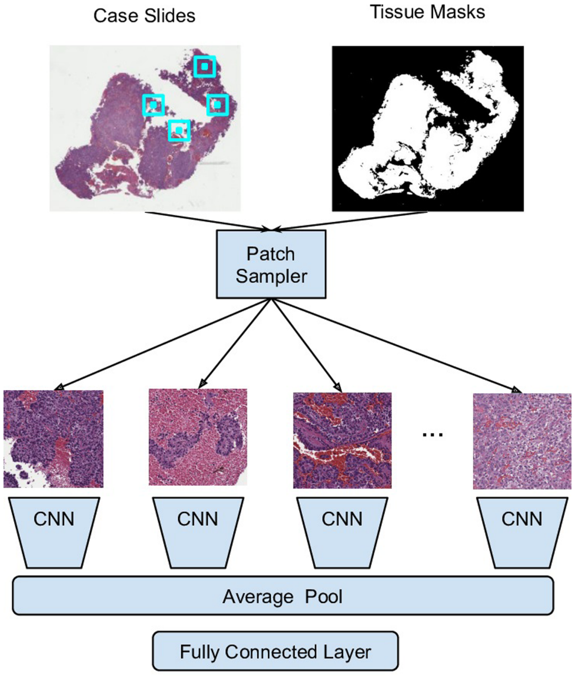

Figure 1: The illustration of image patching technique. Image taken from [7]

In fact there are more approaches of handling the whole slide images by patching giving good results [12] [32] and proving it to be an effective way. The models range from simple CNNs to more complicated segmentation models identifying some cancer type cells. The neighboring patch information is tried to be incorporated by using CNN - LSTM architecture [1] or CRF (Conditional random fields) [18]. Training on image patches still remain a time consuming process, so using visual-attention models were proposed to optimize it [25].

2 Methodology

In this chapter we are going to present the relevant methods and definitions that are related to our research. Our modeling consists of combining convolutional neural network (CNN) results with survival analysis. So we start with survival function material followed by the main concepts and functionalities of deep neural networks. Training CNN models is a time consuming process requiring a specific way of handling - we cover this in another chapter.

2.1 Survival analysis

Survival analysis is statistical branch that analyzes the expected duration of time until an event of interest happens. Regarding patient survival the main questions that are analyzed are: What population part survives past a certain time? After surviving up to current moment what is the rate of dying? How and what characteristics impact the probability of survival? Survival question may also be formulated as a recovery question. Besides the topic of death in biological organisms, this branch is also applied in other fields, such as event prediction in economics, sociology, epidemiology or other. Some popular examples would be the next purchase in retail, time to a product/mechanism or system failure.

In general, survival function is the probability that the time of death is later than some specified time t and is defined as follows

$$S(t) = P(T > t) \tag{1}$$

There are a few approaches of survival analysis regarding different target and specifics of data. In this thesis we are going to use:

- Kaplan–Meier curves for survival times of members of a group.
- Cox proportional hazards regression to analyze the effect of categorical or quantitative covariates on survival.

A common problem in this field of statistics is censoring - missing data problem when time to event is not observed properly.

2.1.1 Survival and hazard functions

Lets denote $T_1, T_2, ...T_n$ - random discrete positive variables meaning time of event. Then having

- $F(t) = P(T \leq t)$ - cumulative distribution function,
- $f(t)$ - density function,
- $S(t) = 1 - F(t)$ - survival function,

we define a hazard function:

$$\lambda(t) = f(t)/S(t) \tag{2}$$

and cumulative hazard function

$$\Lambda(t) = -ln(S(t)) \tag{3}$$

2.1.2 Data censoring

There are three types of data censoring:

- Right censoring in which a certain event does not happen during the time observed, rather may be expected to happen later.
- Left censoring is when the event of interest starts before the start of study.
- Interval censoring is when the time of event is known to happen in certain time interval.

In this thesis we are going to work with data of random right-censoring. Practically, this may be caused in case of termination of study before the event of interest, for example, a patient leaves to another hospital, or simply shows up no more.

Right-censoring definition.

Supposing that the moment of death (or other event) T_i is known only if $T_i \leq C_i$, where $C_i \geq 0$ is a random value or a constant. Otherwise, the value of random variable T_i is unknown, but it is known to be bigger than the censoring value C_i. C_i is called a right-censoring moment.

Lets denote $X_i = T_i \wedge C_i, \delta_i = 1_{T_i \leq C_i}$. Here δ_i is an indicator obtaining value 1 if T_i is known and 0 otherwise. Set of pairs

$$(X_1, \delta_1), (X_2, \delta_2), ..., (X_n, \delta_n) \tag{4}$$

is called a right-censored sample. Here $X_i = min(T_i, C_i)$ - time to event, T_i - failure time, C_i - censoring time, $\delta_i = 1_{T_i \leq C_i}$ - event censoring indicator.

In case of C_i being a random variable the sample is random right-censored.

2.1.3 Parametric, semi-parametric and non-parametric models

In terms of parametric nature the models may be of 3 types: parametric, semi-parametric and non-parametric.

Statistical model $P = \{P_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ is parametric if $\boldsymbol{\theta}$ is finite-dimension:

$$\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_m)^T \in \boldsymbol{\Theta} \subset \boldsymbol{R}^m \tag{5}$$

Model $P = \boldsymbol{P_{\theta}}, \boldsymbol{\theta} \in \boldsymbol{\Theta}$ is non-parametric if $\boldsymbol{\theta}$ is infinite-dimension and does not have finite-dimension components.

Model is semi-parametric if $\boldsymbol{\theta}$ has both finite-dimensional and infinite-dimensional components.

In this work we are going to use Cox Proportional Hazards Regression which belongs to the class of semi-parametric models, and Kaplan-Meier which is non-parametric.

2.1.4 Maximum likelihood method for right-censored sample

Maximum likelihood method is a frequently used method for the search of the best parameter estimate set.

Suppose we have a sample $X \sim f(x, \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \boldsymbol{R}^m$. Lets define $L(\boldsymbol{\theta}) = f(x, \boldsymbol{\theta})$ - a likelihood function.

$\hat{\boldsymbol{\theta}}$ is called maximum likelihood estimator if

$$L(\hat{\boldsymbol{\theta}}) = max_{\theta \in \boldsymbol{\Theta}} L(\boldsymbol{\theta}) \tag{6}$$

The log-likelihood is often used:

$$l(\boldsymbol{\theta}) = \ln L(\boldsymbol{\theta}) = max_{\theta \in \boldsymbol{\Theta}} L(\boldsymbol{\theta}) \tag{7}$$

ML function value is calculated by the given formula:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} p(X_i, \boldsymbol{\theta}) \tag{8}$$

where $X = (X_1, X_2, ...X_n)^T$ is a simple random sample and $X_i \sim p(s, \boldsymbol{\theta})$.

Analogically for the log-likelihood:

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p(X_i, \boldsymbol{\theta}) \tag{9}$$

In case of right censoring, suppose that censoring times C_i do not depend on parameters $\boldsymbol{\theta}$, ML function is

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \lambda^{\delta_i}(X_i, \boldsymbol{\theta}) S(X_i, \boldsymbol{\theta}) \tag{10}$$

Then log-likelihood can be defined as

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} (\delta_i \cdot ln\lambda(X_i, \boldsymbol{\theta}) + lnS(X_i, \boldsymbol{\theta})) \tag{11}$$

To sum up searching for the best parameter estimates $\hat{\boldsymbol{\theta}}$ the aim is to maximize the likelihood (or log-likelihood) function.

2.1.5 Kaplan-Meyer estimate

Kaplan-Meyer estimate is a non parametric estimate of survival function. Its important advantage is that the method can handle right censored data. It is defined as

$$\hat{S}(t) \equiv \prod_{i, t_i \leq t} \frac{n_j - d_j}{n_j} \tag{12}$$

where t_i - time when at least one event happened, d_j is a number of events at t_i, n_i - the number alive, and at risk, just before time t_i.

The Kaplan–Meier estimate is one of the most popular methods of survival analysis. It analyzes recovery/death rates, may work for measure effectiveness of treatment. However, this method is univariate. It can be applied in certain observation group, defined by a single categorical variable or combination them, but is not designed to investigate the other explanatory covariate (especially quantitative) associations with the survival outcome. The impact of explanatory covariates is estimated using Cox proportional hazards model.

2.1.6 Cox proportional hazards regression

Cox proportional hazards regression is a semi parametric model. It is very frequently used approach for investigation of the survival dependence on explanatory covariates, both quantitative and categorical variables.

Cox proportional hazards regression models the hazard function. Hazard simply spoken is a risk of dying at time t. It can be estimated by a given formula:

$$\lambda(t,x) = e^{\beta_1 x_1 + \cdots + \beta_k x_k} \lambda_0(t) \tag{13}$$

where $\lambda(t,x)$ - hazard function, $\lambda(t)$ - baseline hazard function, $(x_1, x_2, ..., x_k)$ - covariates constant over time, $\beta = (\beta_1, \beta_2, ..., \beta_k)$ - coefficients, of corresponding covariates.

Cox proportional hazards regression may be used assumed that the hazard curves for the patients is proportional, that is for any two observations with covariates $x^{(i)}, x^{(j)}$ the ratio of hazard rates is constant over time.

$$\frac{\lambda(t, x^{(i)})}{\lambda(t, x^{(j)})} = e^{\beta^T (x^{(i)} - x^{(j)})} \tag{14}$$

The estimate $\hat{\beta}$ is found by maximizing the partial log-likelihood function.

Lets denote our data

$$(X_1, \delta_1, x^{(1)}), (X_2, \delta_2, x^{(2)}), ..., (X_n, \delta_n, , x^{(n)}) \tag{15}$$

where X_i - event of censoring time, δ_i - censoring indicator, $x^{(i)}$ - explanatory variables (covariates).

Then the partial log-likelihood function

$$l(\beta) = \sum_{i=1}^{n} \delta_i (\beta^T x^{(i)} - \sum_{i=1}^{n} ln(\sum_{j:X_j >= X_i} e^{\beta^T x^{(j)}}) \tag{16}$$

Checking Cox proportional hazard assumptions

Checking if the data is suitable for Cox proportional hazard regression analysis Schoenfeld residuals are calculated.

Firstly we define weighted empirical covariance matrix of the covariate values for all individuals at risk (non-censored and not failed) just prior time X_i:

$$V(X_i, \hat{\beta}) = \frac{\sum_{l:X_l \geq X_i} (z^{(l)} - E(X_i, \hat{\beta}))(z^{(l)} - E(X_i, \hat{\beta}))^T e^{\beta^T} z^{(l)}}{\sum_{l:X_l \geq X_i} e^{\beta^T} z^{(l)}} \tag{17}$$

The Schoenfeld residuals are the difference between observed and expected covariate values at each failure time:

$$\hat{s_i} = \delta_i (z^{(i)} - E(X_i, \hat{\beta})) \tag{18}$$

here X_i is failure time of i^{th} observation, $z^{(i)}$ - i^{th} vector of covariates, δ_i - censoring.

Then scaled Schoenfeld residuals are defined as:

$$\hat{s}_i^* = V^{-1}(X_i, \hat{\beta}) \hat{s}_i \tag{19}$$

To qualify for Cox regression all the components of these residuals ($\hat{s}^*_{ij}$) must by dispersed around zero vector (slope=0).

We will formulate the test of the hypothesis that the data meets the proportional hazard assumption, or the slope of Schoenfeld residuals is 0.

Suppose we have a linear function $\hat{s}^*_{ij} = \rho_j(g(X_j) - \hat{g}) + \epsilon_i$, where g - time function, $\hat{g} = \sum_{i:\delta_i=1} g(X_i)/\delta$. $\hat{s}^*_{ij} \sim 0$ equivalently meaning that $\rho_j \sim 0$. The estimator of ρ_j and the test statistic T is defined as follows:

$$\hat{\rho}_j = \frac{\sum_{i:\delta_i=1}(g(X_j) - \hat{g})\hat{s}^*_{ij}}{\sum_{i:\delta_i=1}(g(X_j) - \hat{g})^2} \tag{20}$$

$$T_j = \frac{(\sum_{i:\delta_i=1}(g(X_j) - \hat{g})\hat{s}^*_{ij})^2}{\delta \hat{I}^{jj} \sum_{i:\delta_i=1}(g(X_j) - \hat{g})^2} \tag{21}$$

where $\hat{I}^{jj}$ - estimated variance of $\hat{s}^*_{ij}$. The hypothesis that proportional hazard assumption is satisfied (H_0) the distribution of statistic T_j is approximately χ^2 with 1 df. If H_0 is rejected it means that the covariate violates of the PH assumption.

2.1.7 Interpreting Cox proportional hazard regression results

- Covariate significance Cox proportional hazard regression estimates the significance of the covariate x_i by setting a hypothesis $H_0 : \beta_i = 0$, where β_i is a corresponding parameter of x_i. The hypothesis is rejected if p-value is lesser than selected significance level: $p < 0.05$ indicating particular variables significance.

- Akaike information criterion (AIC) is a prediction error estimator showing model information loss. The lower value indicates the better model.

  $$AIC = -2(\hat{L}) + 2k \tag{22}$$

  where $\hat{L}$ - estimated value of likelihood function, k - number of parameters.

- Concordance index - the concordance index is a metric to evaluate models performance. It measures the proportion of concordant pairs divided by the total number of possible evaluation pairs.

  Concordance is computed by the following formula:

  $$C = \frac{\sum_{i,j} 1_{T_j < T_i} \cdot 1_{\eta_j < \eta_i} \cdot \delta_j}{\sum_{i,j} 1_{T_j < T_i} \cdot \delta_j} \tag{23}$$

  where η_j - j^{th} patients risk score, T_j - time of event or censoring, δ_j - censoring indicator.

  The model is considered better if the concordance index is closer to 1; the value of 0.5 indicates random guessing so the model is poor.

- Hazard rates

  e^{β_i} - hazard rate measuring the impact of a covariate x_i when it is changed by one unit of measurement. The conclusions can be made that

- if $hazardrate > 1$ the corresponding covariate increases the survival time,
- if $hazardrate = 1$ the corresponding covariate does not affect survival time,
- if $hazardrate < 1$ the corresponding covariate decreases the survival time.

2.1.8 Selection of covariates

A common problem having a large number of covariates is to select those of highest significance. In the thesis we use two ways to solve it.

Forward (step up) variable selection

The forward (step up) variable selection procedure is well suitable when having multi-colinearity. The algorithm is defined like this:

1. Initially there are no variables in the model.
2. Check p value of all covariates not yet in the model adding one by one to the current model.
3. Update the current model by adding the new significant covariate with lowest p-value (having significance level set to 0.05).
4. Continue adding new covariates until no new significant ones are found.

Investigation of correlation matrix

To analyse the correlation between variables often a correlation matrix is used. This matrix simply includes pairwise correlation coefficients between variables. The dimension of this matrix is n x n, where n is a number of covariates. A correlation coefficient obtain values from interval [-1, 1]. There is no relation between two variables if the coefficient = 0, and the relation is stronger (negative or possitive) when the values are closer to interval ends (-1 or 1).

2.2 Artificial neural network

Artificial neural networks (ANN) - one of supervised machine learning algorithms.

Supervised learning algorithms are used for data sets having some ground truth or target information, whether it is categorical or quantitative. Data with categorical targets are modelled by classification models, quantitative (having values from a continuous interval) by regression models.

Logic behind ANN is similar to brain: it consists of many neurons which are interacting and processing the input signal and producing the output.

These neurons are distributed in so called layers. The process begins with an input layer which in other words is a certain data analysed. The following hidden layer processes the input layer and produces the output that goes as an input for the next layer. At the end there is an output layer, that produces calculated results in our yielded form.

The output of the ANN is then compared to the ground truth (the known correct output) using a certain loss function measuring the "distance" between the predicted and true values. This function depends on the nature of the target to be predicted.

By the process of backwards propagation [17] the current parameter estimates (a.k.a weights) are being recalculated until minimizing the loss function.

A neuron is defined as follows:

$$y = \phi(\sum_{j=1}^{m}(w_j \cdot x_j)) \tag{24}$$

where m is the dimension of input, x_j - j^{th} input of a neuron, w_j - corresponding weight, and ϕ - non-linear activation function.

2.2.1 Activation functions

In general, activation function is a function that determines the output of a layer. It is defined as follows:

$$Z = f(X) \tag{25}$$

where X is some layer output matrix, Z - activation matrix of same dimensions as X. There are natural restrictions to the activation functions. They must be non-linear, allowing the model to create complex mappings between layers, thus enabling it to learn more complex features. Activation functions must be computationally inexpensive, because they are calculated across all neurons for each data unit. Neural networks may have from thousands to millions of parameters. Therefore, the most popular activation functions have settled:

1. Rectified Linear Units (ReLU).

$$ReLU(x) = max(0, x) \tag{26}$$

This activation may sometimes lead to a so called vanishing gradient problem, when inputs approach zero, or are negative. The gradient of the function becomes zero, the network can no longer learn. The modification Leaky ReLU (LReLU) with any small slope ϵ is an alternative that solves this problem.

$$LReLU = max(\epsilon x, x) \tag{27}$$

2. Sigmoid

$$S(\hat{y}) = \frac{1}{1 + e^{-\hat{y}}} \tag{28}$$

3. Softmax

$$\sigma(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_j^k e^{\hat{y}_j}} \tag{29}$$

where $\hat{y}_i$ is the i^{th} component of predicted $\hat{y}$, k - number of $\hat{y}$ components.

Sigmoid and softmax are typically used only for the output layer. Their advantage is the capability to scale the output components in the interval[0,1] and transform it into a probabilistic form. Sigmoid is used for binary classification, softmax - categorical classification.

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) share most concepts like ANN with slightly altered neuron "communicating" system. Instead of fully connected networks, when each neuron in one layer is connected to all neurons in the next layer, CNNs take advantage of hierarchical patterns and use smaller sets of weights, that are repeatedly applied throughout input while calculating the output [9].

Convolutional neural network specializes in more effective modeling of spatially arranged and high dimensional data, therefore widely applied in image analysis [9].

Next we will define main concepts of CNN.

2.3.1 Convolutional layer

Most part of CNN consist of convolutional layers that convolve producing a dot product of weight and input matrices. This kind of structure enables checking for features in smaller dimension patches. A single convolution has similar form as a neuron. The definition is:

$$y = \phi(W * X_{ij}) \tag{30}$$

where X_{ij} - input matrix patch of some dimension (usually small, for example 3x3, 5x5) from "location ij",W - weight matrix, of the same dimensions as input patch and ϕ - activation function.

A single convolution weight matrix is called a filter. Filters corresponding to a certain layer produce a matrix called a kernel. Convolutional layers transform input matrix to output matrix by applying the convolutional kernel.

Suggesting by name, a filter works as some feature detector, therefore kernel identifies a set of occurring patterns in data. This information is passed on to the next layer resulting in more complicated pattern detection at further layers. The earlier defined artificial neural networks (ANN) tend to increase the number of parameters very fast along with increased data dimensions. This is caused by consisting only of fully connected layers. On the other hand CNNs are more suitable for image analysis not only because of the mentioned pattern recognition nature but also the "economic" efficiency by reusing the small amount of parameters in the filter throughout the input grid, maintaining number of parameters even if the input is large.

From practical point of view, the early layers learn to detect simple elements, such as lines, corners, squares, whereas the latest layers may detect more complex structures enabling a meaningful output [9].

2.3.2 Pooling layer

Another layer specific to CNN is pooling. It works as a filter, which does not include weights rather than a simple rule or function that manipulates the input matrix patch outputting a single value. Pooling layers are used for reducing the spatial size of the representation also causing decrease of the number of parameters and computation in the network. On the other hand, lots of valuable information is lost.

2.4 Deep neural network model architectures / architecture blocks

There exist various deep CNN model architectures, based on different logic applied for separate blocks or even to the whole architecture. In this thesis a convolutional autoencoder and classifier is used, therefore only the related model concepts are given.

2.4.1 Classification models

Classification models are the most common and conditionally simple. The structure consists of enough layers to extract meaningful information and the final output classification layer with the number of neurons equal to different classes our data may obtain. As mentioned we use CNN. The part of layers until the output is called encoder. Suggested by name, here the meaningful features are transformed (encoded). The last layer of the encoder is usually called a bottleneck, the features accordingly - bottleneck features. Finally the bottleneck features are combined to output class predictions see Figure 2.

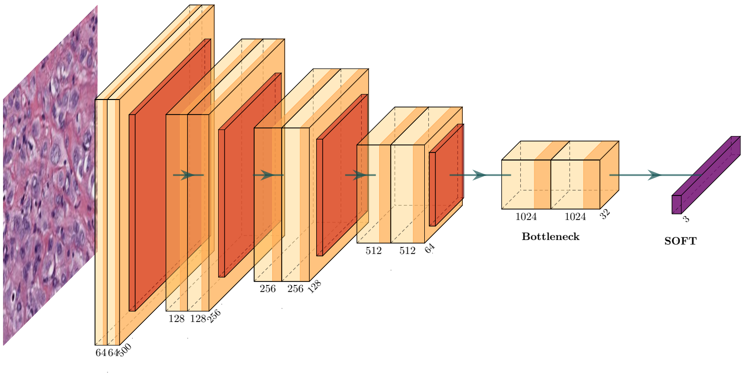

Figure 2: Illustration of CNN model for classification

Depending on the number of output classes, the fully connected layer is added followed by a suitable activation function.

Loss function used for training a classification CNN is cross-entropy, defined as:

$$L(y, \hat{y}) = -\sum_{i=1}^{n} y_i \ln \hat{y}_i \tag{31}$$

2.4.2 Segmentation models

Instead of classifying images, the aim of segmentation models is to classify every pixel in the image. Therefore different structure of model is used. Besides the encoder part segmentation models also have the decoder. Encoders decrease the size (downsample) of the activation maps while decoders are supposed to increase the dimensions (upsample) back to original size. Finally the output layer is added. Segmentation model produces the classification output for each pixel.

A common example of a segmentation model architecture is U-net [26].

2.4.3 Autoencoders

An autoencoder is a type of artificial neural network used to learn a representation (encoding) for a set of data, typically for dimensionality reduction or denoising. In terms of structure it is very similar to a segmentation model see Figure 3.

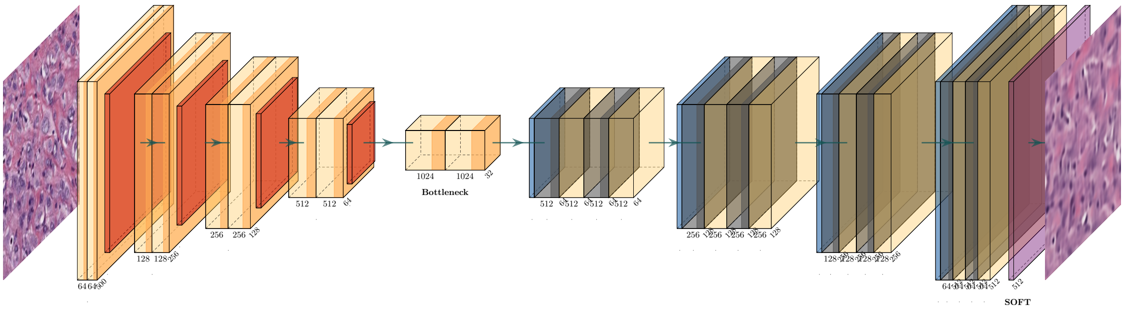

Figure 3: Illustration of model used for autoencoder

The two key points of autoencoder:

- Having some image for an input the ground truth is the same image.
- Depending on the purpose of the autoencoder the bottleneck size is adjusted.

In the thesis we are going to use an autoencoder to transform original image into small dimensional representation by making reasonably "narrow" bottleneck.

Loss functions used for autoencoders:

- Mean Square Error (MSE) defined as:

$$MSE(y, \hat{y}) = \frac{1}{N} \sum (y - \hat{y})^2 \tag{32}$$

  where N - number of pixels in image.

- Structural similarity index (SSIM), defined as:

$$SSIM(y, \hat{y}) = l(y, \hat{y}) \cdot c(y, \hat{y}) \cdot s(y, \hat{y}), \tag{33}$$

  where

$$l(y, \hat{y}) = \frac{2\mu_y \mu_{\hat{y}} + \epsilon_1}{\mu_y^2 + \mu_{\hat{y}}^2 + \epsilon_1} \tag{34}$$

$$c(y, \hat{y}) = \frac{2\sigma_y \sigma_{\hat{y}} + \epsilon_2}{\sigma_y^2 + \sigma_{\hat{y}}^2 + \epsilon_2} \tag{35}$$

$$s(y, \hat{y}) = \frac{2\sigma_{y\hat{y}} + \epsilon_3}{\sigma_y + \sigma_{\hat{y}} + \epsilon_3} \tag{36}$$

ϵ is a small quantity to avoid the computing errors.

2.5 Data transformations

With usage of powerful deep neural networks the problem of insufficient data emerged. Therefore a common approach to enlarge the number of images to train on is image augmentation. There are some initial transformations used to generate slightly different but remained informative images:

- horizontal and vertical flip,
- zoom in/out,
- rotation at random angle,
- color manipulations (darken/brighten, modifying contrast),
- generating random occlusions,
- distortions and other

Besides generating more training data, augmentation also partly prevents model from overfitting the training data.

Data normalization

Image pixel values originally are in range [0,255]. For better generalization they are transformed into interval [0,1].

$$I_N = \frac{I - I_{min}}{I_{max} - I_{min}} \tag{37}$$

here I represents one channel of an image. In case of 3 channels (RGB) each channel is normalized.

3 Practical part

3.1 Introduction of TCGA data specifics

One of the most relevant initiatives to empower the research was National Cancer Institutes "The Cancer Genome Atlas Program" [1]. It provides over 20,000 molecularly characterized cancer cases of 33 cancer types. TCGA generated over 2.5 petabytes of data. This effort began in 2006 and its publicly available database is still the most significant and popular cancer investigation object. In our thesis we are going to use data from this database.

Besides the case information fixed variables, like persons age, gender, tumor type, etc., TCGA project provides large database of histopathology images tissue and diagnostic slides. More details about the preparation of slides are given in the following paragraphs to ground the choice of selected research data.

3.1.1 Preparation of bio-specimen images - slides

Firstly, after biopsy a technician prepares a slide before examining the tissue. The specimen is cut into thin slices. These slices are stained with various dyes, which make the part of cells clearer. They are put on a glass slide and analyzed under a microscope. There are two types of slides, regarding the slide preparation method.

- Permanent section/tissue

  In this case the so called Formalin Fixed Paraffin Embedded (FFPE) tissue is produced in order to save it unchanged for long time. First the specialist places the tissue into formaline for a few hours, then the water is removed from it and replaced by parafin wax. After the paraffin block hardens, the biospecimen is cut into extremely thin slices. The tissue is also dyed in the process. The nucleus of the cell is dyed dark blue and the cytoplasm is pink. Finally the photo is taken from slides.

- Frozen section/tissue

  In this case after biopsy the specimen is quickly frozen. It is then cut into thin layers. These slices are placed on the slide and stained as in a fixative tissue. Then the slides can be made by taking photos of these slices see Figure 4.

The slides are also of two types in different aspect: diagnostic slides and tissue slides.

- Diagnostic slides are aimed for diagnosing a patient from a tissue image and are often done from FFPE slices. The quality of these slides is often much higher as it they supposed for image analysis. Diagnostic slides originate from the same tumor as tissue slides, unfortunately their relationship to the material submitted for genomic analysis is unknown therefore less precisely explaining one another. For example, larger tumors may not be homogeneous,

[1] https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

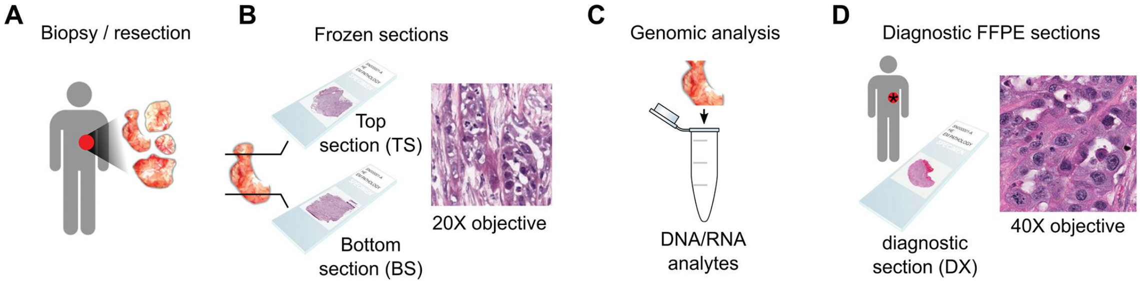

Figure 4: Illustration of tissue usage. Image taken from [5]

and it is not always clear where the frozen tissue was sampled relatively to the diagnostic slide tissue.

- On the other hand the frozen sections do represent the tissue genomic signatures. Tissue slides are often from frozen slices and are used as an additional information. In general frozen tissues are aimed for molecular/chemical analysis. In our data if such is available it is derived indeed from the frozen tissue. However these slides are noted as not quite suitable for image analysis because of the freezing artifacts.

So diagnostic slides carry more details, and have sufficient quality to confirm diagnosis, but the molecular/chemical information is better represented in tissue slides. This trade-off is important in planning image analysis. Since this thesis focuses on whether and how image data can be employed for survival analysis, we chose analyzing diagnostic slides.

3.2 Selected data

In this thesis only one type of cancer data is selected for analysis - Breast cancer. There are 1119 images from 1048 patients (cases) in this group (since some of the patients had more than one diagnostic slide provided - usually not more than 2). In our research we are not going to analyze the relations between two slides of the same patient, therefore we will select one image for case.

3.2.1 Fixed case / sample variables

After rejecting completely uninformative case variables (those not having values, or being the same for each case) 13 variables remain:

- age_ at_ index - age at diagnosis time,
- gender - gender,
- race - race,
- ethnicity - ethnicity,
- tumor_ stage - tumor stage from 1 to 4, or X - unknown,

- ajcc_ t - main (primary) tumor size from 1 to 4, or X - unknown,
- ajcc_ n - level of cancer spreading to nearby lymph nodes, from 0 to 3 or X.
- ajcc_ m - level of metastasis spread, obtaining values 0,1 or X
- histologic_ diagnosis - tissue related cancer type,
- tumor_ sample_ weight - weight of the tumor sample,
- tumor_ nuclei_ percent - percent of diseased cells (the estimation is made by a professional from tissue slides - not diagnostic slides),
- necrosis_ percent - percent of dead cancer cells in the sample (estimated from tissue slides),
- laterality - left or right breast

3.2.2 Data preparation for base model

Initial data table preparation:

- age variable is separated into age groups maintaining a sufficient number of observations. The new variable categorical variable "agegroup" is created.
- Categorical variables are converted into dummy variables

agegroup [year bounds]	Number of obs.
ag_1 (0,45]	181
ag_2 (45,55]	264
ag_3 (55,65]	295
ag_4 (65,75]	183
ag_5 (75,90]	125

Table 1: Distribution of observations by age group

In the next step most variables are removed because of not being identified as significant. To select significant variables we use the forward variable selection method (with Cox regression).

3.3 Cox regression model

The two significant variabes remain: agegroup and tumor stage. According to the standards of cancer stage reporting [2], the T, N, M indicators carry similar information as stage since these are two different systems of measuring the disease. We shall as well try using ajcc_t, ajcc_n, ajcc_m. Here we list the dummy variables. Note, that the first level of each group is removed as a base class.

[2] https://www.cancer.gov/about-cancer/diagnosis-staging/staging

1. Group ajcc_m dummy variables: m1 - metastasis level 1. m0 (no metastasis found) removed.

2. ajcc_n dummy variables: n1 - level 1, n2 - level2, n3 - level 3. n0 (no tumor cells near lymph found) removed.

3. ajcc_t dummy variables: t2 - level 2, t3 - level 3, t4 - level 4. t1 (tumor size of level 1) removed.

4. tumor_stage dummy variables: levels s2, s3, s4 left, s1 (stage 1) removed.

5. agegroup dummy variables: a1 (age up to 45 years) removed, others left (1). Levels note the intensity of a certain factor.

The significance of variables may be unexposed in a single model because they are obviously correlated. However the Cox regression model does not require the covariates to be independent, so we will compare two sets of base variables and select the better model.

1. Variables: ajcc_t, ajcc_n, ajcc_m and agegroup

covariate	exp(coef)	exp(coef) lower 95%	exp(coef) upper 95%	p-value
ag_2	1.17	0.66	2.08	0.58
ag_3	1.18	0.66	2.12	0.58
ag_4	2.06	1.13	3.75	0.02
ag_5	4.82	2.6	8.76	<0.005
m1	3.37	1.65	6.87	<0.005
n1	1.71	1.10	2.65	<0.005
n2	2.84	1.61	5.00	<0.005
n3	2.42	1.10	5.344	0.03
t2	1.41	0.86	2.295	0.17
t3	1.38	0.72	2.62	0.33
t4	2.39	1.12	5.10	0.03

Table 2: Initial model with variables: agegroup, t, n, m

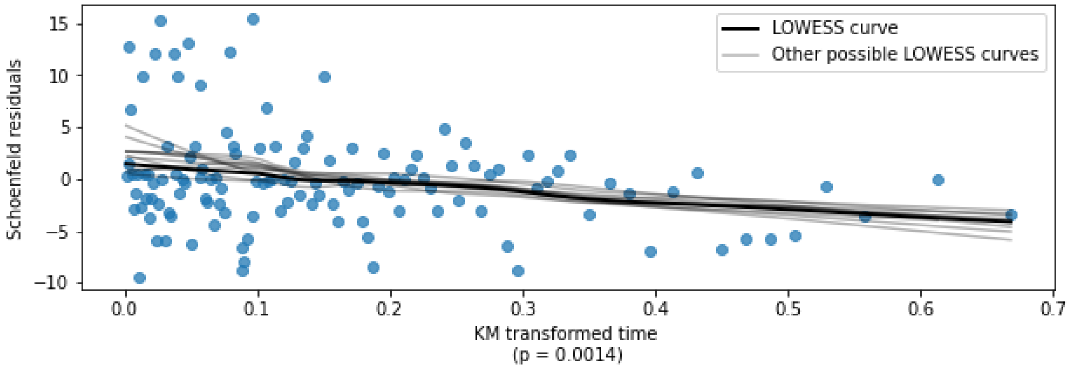

Figure 5: Schoenfeld residuals of covariate t4

The initial Cox regression results show that covariate t4 does not meet the proportional hazard assumptions (p value < 0.005). The Schoenefeld residuals of t4 for Kaplan-Meier estimate are shown in figure 5. However none of t2, t3, t4 are significant, so we remove them from the model.

We check the Cox regression results for remaining variables. Results are in Table 3. Partial AIC = 1337, Concordance = 0.72.

covariate	exp(coef)	exp(coef) lower 95%	exp(coef) upper 95%	p-value
ag_2	1.15	0.65	2.04	0.62
ag_3	1.14	0.63	2.04	0.67
ag_4	2.18	1.21	3.93	0.01
ag_5	4.56	2.53	8.22	<0.005
m1	3.91	1.96	7.83	<0.005
n1	1.97	1.29	2.99	<0.005
n2	3.18	1.83	5.50	<0.005
n3	3.00	1.45	6.23	<0.005

Table 3: Final Cox regression results for variables: agegroup, m and n

Here all the covariates satisfy the proportional hazard assumption (critical value of p is 0.05).

2. Variables: tumor stage and agegroup

The initial Cox regression model shows that variable s3 violates the proportional hazard assumption - p value < 0.005. The residuals are shown in Figure 6. In this case, variable s4 is significant so we stratify it.

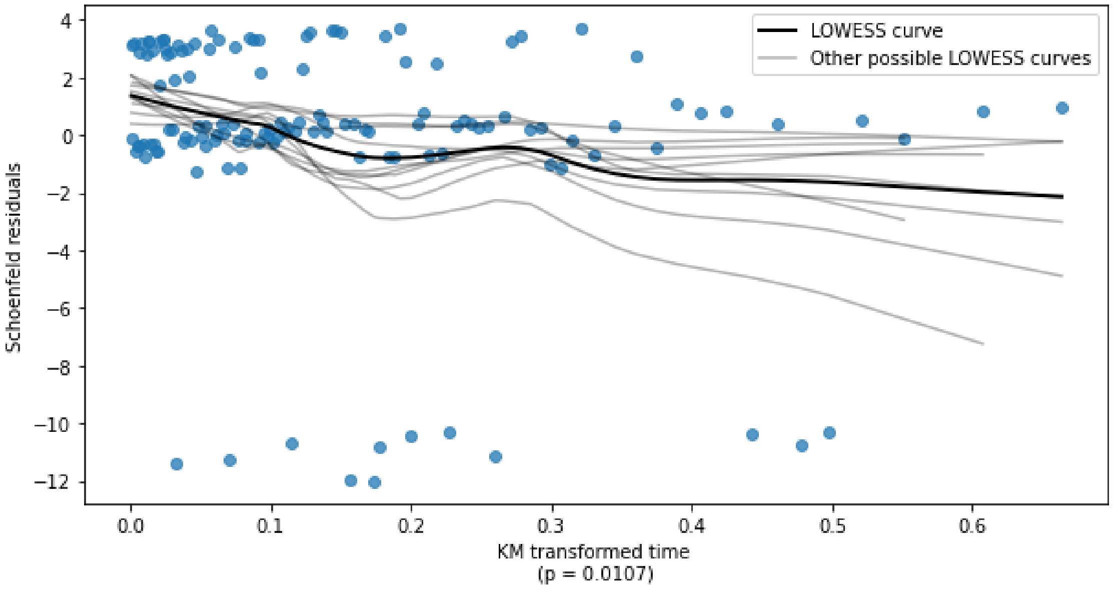

Figure 6: Schoenfeld residuals of covariate s3

The final results of Cox regression are better than of the first model, partial AIC = 1270, Concordance = 0.73. So we select this one see Table 4 as a

covariate	exp(coef)	exp(coef) lower 95%	exp(coef) upper 95%	p
ag_2	1.06	0.60	1.88	0.83
ag_3	1.06	0.60	1.89	0.85
ag_4	2.12	1.20	3.76	0.01
ag_5	4.29	2.41	7.66	<0.005
s2	2.35	1.27	4.37	0.01
s4	17.31	7.76	38.62	<0.005

Table 4: Final Cox regression results for variables: agegroup and tumor stage

> base model for further investigation. We see that belonging to age groups ag_2,ag_3,ag_4,ag_5 increases the risk of death correspondingly: 1.06, 1.06, 2.12 and 4.29 times. Belonging to stage group 2 and 4 increases it 2.35 and 17.31 times.

Belonging to age groups 2 and 3 gives very similar increase which is seen in the estimated Kaplan-Meier curves by age plotted in Figure 7. The plots also illustrate the stratification of s3. By stratifying the variable s3, we get two different baseline hazard functions and therefore two sets of curves.

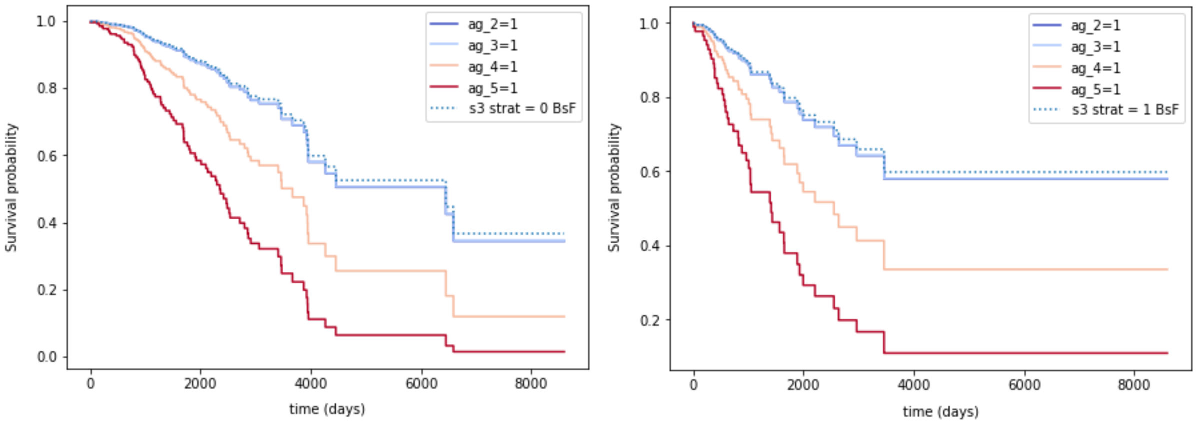

Figure 7: Kaplan-Meier curves by age. On the left there is group s3=0, on the right s3=1

3.4 Image processing

3.4.1 Image patching

Large whole slide images (WSI) are processed as follows.

- Image is resized 16 times, by reducing a single dimension 4 times.
- Resulted image is divided into grid of patches (256pixels x 256pixels). Note that every whole image patch number is not equal, it depends on the tumor area pictured, however scale (zooming) is the same.

- Uninformative (empty/noisy) patches are removed.

  1. Firstly, we know for sure that empty patches must be removed, we only analyze a tissue.
  2. Second, there are some gray darker regions, spots - simply photography artifacts. They carry no useful information.
  3. Third, images sometimes tend to have markings which are done by specialist. They are often done with an red, blue or green markers. Patches including intensive marking often lose the information - the cells are not visible or the color scheme becomes completely different, so we chose to remove those too.

A fast and efficient enough rule was selected: First, the mean RGB vector is calculated. If standard deviation (std) of this vector is less than 3 – that indicates a color close to gray. In the image it filters the empty or photo noisy patches. If std > 70 it indicates too saturated color, which catches sometimes appearing marker lines, especially those painted not on the tissue part see Figure 8.

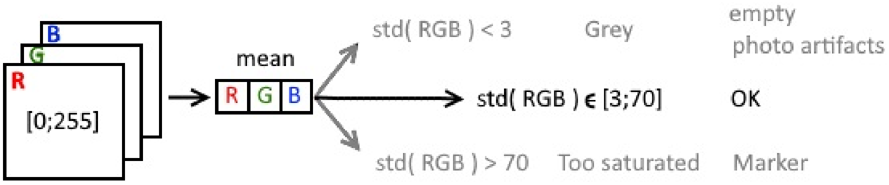

Figure 8: Illustration of informative image filtering procedure

By this simple method we removed only truly irrelevant patches - those that did not have any tissue in them. Some mentioned practices from related works highlight the importance of informative patch selection. It is usually applied in more concrete tasks searching for some defined cell elements, however yet there is no reliable information about the regions that affect survival. Therefore, our analysis runs on more than 3.5 million of remaining image patches.

3.5 Convolutional models

Our aim is to investigate the extraction meaningful information from images. The idea is by using convolutional neural networks:

- transform an image into lower dimensional space
- extract abstract composite features of a whole image

Two types of models are used: an autoencoder and a classifier. We focus on approaches of feature extraction from images minimizing the demand of resources:

- The models are run on image patches separately without taking neighboring information into account.
- Training includes partial data set of representative whole images. The complete data set is used for final image feature template generation.

3.5.1 Bottleneck feature processing framework

After model shows reasonable training results the set of parameters are fixated. Using these parameters the bottleneck vectors are generated from each patch. Then two sets of aggregated whole image representations are produced by calculating the average (lets call it "avgvector" for future reference) and maximum value vector ("maxvector") along the patches of certain whole image.

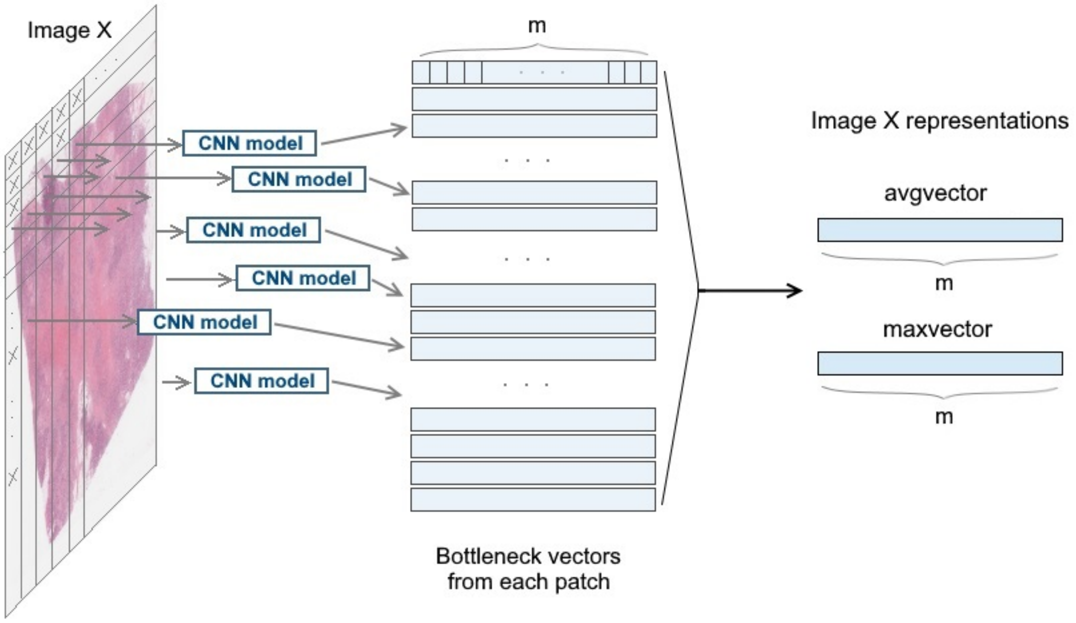

Figure 9: The scheme shows the process generating whole image lower dimensional representations. Note that due to the schematic simplicity the number of patches is shown smaller (in reality it varies from 1000 to 10000). Here m - number of bottleneck layer output values

The number of bottleneck features in our models is 512 and 2048. Output high-dimensionality still remains a fundamental problem. In Cox proportional hazards model to avoid over-fitting the recommended ratio of observations and explanatory variables is not less that 10 [31]. Taken 1025 observations into consideration around 100 bottleneck values is a limit.

For future reference lets define some most occurring instances:

- "bottleneck-only" - a table containing only time "T", censoring "delta" and selected bottlenecks
- "bottleneck-full" - a table containing time "T", censoring "delta", the significant fixed variables and selected final significant bottleneck variables.

3.5.2 Autoencoder CNN

Constructing autoencoder the aim is keeping as low bottleneck output dimension as possible maintaining the reasonably good performance, which initially is evaluated by a human observation.

For autoencoder training we use train and validation data sets in order to have better understanding of model generalization capability. The training curves do not show over fitting - training set results slightly get better until some point when the improvement starts getting too slow see Figure 10. On the other hand validation results seem to vary more sometimes.

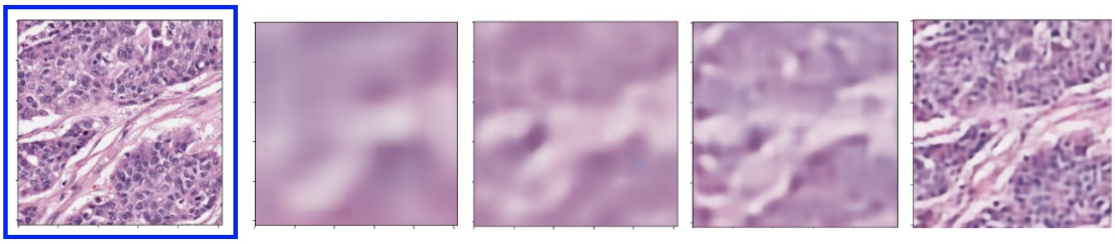

Figure 10: sample output of 4 autoencoders. The first image from the left - original patch

We trained models varying in different bottleneck dimensions. Three loss functions were tried: MSE, SSIM loss and binary cross entropy. Best results by eye were achieved with MSE. For comparison the parametric information and training curves are given for four structurally similar models: autoencoder-64, autoencoder-128, autoencoder-512, autoencoder-2048 (model name includes the number of bottleneck output values).

There is no universal approach to estimate the accuracy of an autoencoder, so the outputs will be compared in the second sections using Cox proportional hazard model. Visually a natural tendency is to have much worse results with decreasing dimensions of bottleneck.

3.5.3 Classifier CNN

We trained a convolutional classifier as a multitask model with 3 clasification ends. It predicted levels of ajcc_m, ajcc_n, ajcc_t. These 3 groups were selected because they are related to the cell structure (therefore we expect their signatures in the images) and they were significant candidates for the basemodel.

Unlike in autoencoder, which had all the necessary information to learn meaningful features, there is no guaranty, that signatures of any selected class is present at a certain patch. This may be called a week supervised learning (Weak supervision in general occurs when data is highly noisy, limited, or imprecise).

We chose to select a balanced sample set of patches (all the image patches from selected whole image are included), regarding similar amounts of occuring classification attributes in each group. This is because of two reasons:

- Classes from each group in the whole data set are highly unbalanced, for example there are only 2% of metastasis level 1.
- There are 3.5 million images, so the duration of training process is beyond our reasonable computing capabilities (one iteration would last about 50 hours), so it is beneficial to narrow down our training data.

A few model architectures were tried and the trends are that the training loss decreases slowly and validation loss is quite unstable. That may indicate overfitting, however in our case it also shows an outcome of a weak supervision. We do not necessarily dismiss the results, because model still learns. The model training is stopped and saved at the best training and validation results.

3.5.4 SurvCNN

Another architecture was tried combining the CNN feature extraction and direct predictor of the survival distribution.

Here we do the training on the same set of patches as in Classifier CNN. This time our targets are times of survival binned into 20 groups/intervals (some experiments have been done with different bin number), therefore the classification ending had 20 nodes and the Logistic Hazard loss was applied. In other words for each time interval the conditional hazard probability is estimated. The loss is given next.

$$L = -\frac{1}{n}\sum_{i=1}^{n}(\delta_i log(h(t_i|x_i)) + (1-\delta_i)log(1-h(t_i|x_i)) + \sum_{j=1}^{k(t_i)-1} log(1-h(\tau_j|x_i))) \quad (38)$$

where t_i - observed event time, δ_i - censoring, h - hazard function, k - index of discrete time t, therefore $t = \tau_{k(t)}$ (or τ_j - time interval), n - number of individuals (cases/images/patients).

3.6 Cox regression results with bottleneck features added

In this section we are going to check hypothesis of the extracted deep feature significance and informativeness. Therefore we are going to investigate the bottleneck-only and bottleneck-full tables with Cox regression.

The steps of analysis

1. Pre-selection of bottleneck features (the number must be reduced to approximately 100) if necessary.

2. Cox regression fitting on bottleneck-only and selection of significant variables for bottleneck-full table.

3. Cox regression fitting on bottleneck-full table.

4. Comparing step 2 and 3 results with the base model.

Notation of bottleneck variables used in the table:

- m_0, m_1, ... - elements of maximal bottleneck feature vector (autoencoder output)

- a_0, a_1, ... - elements of average bottleneck feature vector (autoencoder output)

- cm_0, cm_1, ... - elements of maximal bottleneck feature vector (classifier output)

- ca_0, ca_1, ... - elements of average bottleneck feature vector (classifier output)

3.6.1 Cox regression for autoencoder bottlenecks

As mentioned before our aim is to extract approximately 100 deep features. We will check two models:

- autoencoder-64 - this model has the "narrow" enough bottleneck, that generates 64 variables. It is not exceeding 100 therefore the Cox regression analysis may be applied directly.
- autoencoder-2048 showing the best result. The number of bottleneck variables obtained by this model is 2048.

The advantage of autoencoder-64 is a narrow bottleneck, on the other hand, it produces rather poorly-looking result. By comparing the Cox regression results we will check the hypothesis if direct lower-dimensional output of a poorer model is better than having to pre-select a smaller set of features from high-dimensional output of better model.

autoencoder-64

Cox regression analysis step application.

1. First step is not necessary because bottleneck vector dimensionality is low enough.
2. Cox regression is fitted on Model444 bottlenecks-only table. The analysis show no significant variables for both average and maximal bottleneck vector, high partial AIC (1579) and low concordance index (0.66).

We see that the "poor" model does not provide meaningful data and do not proceed with further analysis and reject this method of feature extraction.

autoencoder-2048

Average bottleneck vector has very high correlations, only 8 covariates with correlations lower than 0.95. Having a bottleneck-only data with these covariate candidates Cox regression shown no significant values. We do not proceed.

Cox regression analysis step application.

1. First we select variables from maxvector by the level of correlation. Correlation threshold 0.7 gives a set of 126 values from the maxvector.
2. Cox regression shows 7 significant variables out of initial 126. Final result of Cox regression with 7 bottleneck variables - autoencoder bottleneck-only model gives partial AIC: 1494 and concordance: 0.63.

covariate	exp(coef)	exp(coef) lower 95%	exp(coef) upper 95%	p-value
ag_2	1.01	0.57	1.80	0.96
ag_3	0.96	0.53	1.71	0.88
ag_4	2.20	1.24	3.91	0.01
ag_5	3.97	2.22	7.12	<0.005
s2	2.28	1.22	4.27	0.01
s4	16.88	7.35	38.77	<0.005
m_15	0.31	0.06	1.59	0.16
m_383	0.07	0.01	1.03	0.05
m_1103	10.25	1.97	54.79	0.01
m_1287	0.20	0.03	1.28	0.09
m_1959	25.65	3.03	217.07	<0.005
m_1991	0.13	0.01	1.54	0.11
m_2031	3.73	0.38	36.65	0.26

Table 5: Bottleneck-full model including deep features from autoencoder

3. the 7 selected significant parameters are included into bottleneck-full table. Cox regression gives 1 significant variable and is run once again of the final bottleneck-full table. The results are given in Table 5.

   Partial AIC: 1262, concordance: 0.76. However after leaving only m_1959 the model does not show them as significant.

4. We see that the bottleneck-only models performance is lower than basemodel - the features extracted do not supersede the basic covariates, however the concordance result is 0.63 showing non random prediction. After combining the significant variables the bottleneck-full model shows no significant bottleneck values, meaning that the information carried by bottlenecks overlap with base variables.

   Also, the results imply that it is better to select a smaller set of features from more promising output (in our case at the beginning there are significant variables found) rather using a poorer models lower-dimensional outputs, that allows to skip the pre-selection phase.

3.6.2 Cox regression for classifier bottlenecks

The number of bottleneck variables obtained by classification model is 512. Here we will repeat the same process of pre-selecting variables by the level of correlation.

Like in the case of autoencoder we witness same trend of avgvector component correlations. There are 53 covariates with correlations lower than 0.95. Hence Cox regression show no significant values. This suggests that average bottlenecks are non informative.

Cox regression analysis step application.

1. We select variables from maxvector by the level of correlation. Threshold 0.8 gives a set of 135 values.

2. Cox regression shows 5 significant variables out of initial 135. Final result of Cox regression with 5 bottleneck variables - classification bottleneck-only model gives partial AIC: 1484 and concordance: 0.66.

3. The 5 significant parameters are included into bottleneck-full table. The results are given in Table 6.

covariate	exp(coef)	exp(coef) lower 95%	exp(coef) upper 95%	p-value
ag_2	1.08	0.61	1.93	0.79
ag_3	1.06	0.59	1.89	0.85
ag_4	2.00	1.13	3.55	0.02
ag_5	3.88	2.14	7.01	<0.005
s2	2.56	1.37	4.81	<0.005
s4	18.37	8.07	41.84	<0.005
cm_4	0.99	0.87	1.12	0.82
cm_123	1.22	1.10	1.36	<0.005
cm_386	0.98	0.89	1.08	0.67
cm_390	1.34	1.11	1.61	<0.005
cm_406	0.92	0.77	1.11	0.40

Table 6: Initial Bottleneck-full model including deep features from classifier

Partial AIC: 1252, Concordance index 0.79. Cox regression gives 2 significant variables: cm_123, cm_390. We rum it once again on the final bottleneck-full table with significant variables left. Result Table 7.

covariate	exp(coef)	exp(coef) lower 95%	exp(coef) upper 95%	p-value
ag_2	1.09	0.61	1.93	0.77
ag_3	1.06	0.60	1.89	0.84
ag_4	2.02	1.14	3.58	0.01
ag_5	3.91	2.18	7.02	<0.005
s2	2.58	1.38	4.83	<0.005
s4	19.45	8.69	43.56	<0.005
cm_123	1.19	1.08	1.32	<0.005
cm_390	1.28	1.09	1.51	<0.005

Table 7: Initial Bottleneck-full model including only final significant deep features from classifier

Partial AIC = 1247, Concordance = 0.78 Here we get a slight lower partial AIC and higher concordance compared to the basis model. cm_123 when increased by one increase the risk of death correspondingly: 1.19 times, variable cm_390 increases it 1.28 times.

4. Bottleneck-only models performance is lower than basemodel as well, however it is higher than in case of autoencoder. Still the features extracted do not supersede the basic covariates, but concordance is suggesting non random

prediction. The bottleneck-full model shows better performance than base-model. Decrease of AIC and concordance higher by 0.5 suggest that classifier maxvector may be used to extract meaningful deep features.

3.6.3 SurvCNN

A set of attempts to train SurvCNN model was run combining different normalization and regularization settings and different encoder architectures (Classifier CNN modifications also Squeezenet [13], other). However none of the trials produced statistically significant bottlenecks for Cox model.

Here the further investigation might go in two directions: a smarter patch sampling/processing or more complex encoder taking the relations between patches into account.

3.7 Computational resources

Working with images and deep neural networks usually computational resources become a matter of consideration. Model training is the most resource-intensive task. Computing process may be parallelized using Graphical Processing Units (GPUs) that have become a common hardware tool in such work. In this thesis NVIDIA GTX 1080-Ti was used.

In this chapter we will discuss the computing related aspects which make groundings of some used approaches.

3.7.1 Data set size information

First of all, the chosen data has 1048 whole slide images (WSI - a common format used in medical imaging) each taking up 1GB space on the average, therefore 1TB on the whole set. Since the training requires smaller images, we performed image patching. Moreover before patching, the original image size was reduced 16 times. In the process there were more than 3.5 million images saved, taking up 64GB, having that size of image is 256 x 256 in pixels.

Note, that the usage of non resized patches would have taken up to one more terabyte of space. Besides, we would either have make 16 times more images, which would make it $\sim$ 56 million image set, either an image would obtain dimensions of 1000 x 1000 in pixels, which would require larger deep neural network models capable to maintain the performance quality.

Having our chosen approach of handling images, part of image patch generation was done in Google Cloud Platform. The data then was transferred to another computing machine with GPU.

3.7.2 Data processing time

Computing time depends on the data set size, neural network model structure and number of parameters. The models used in this research is conditionally low $\sim$ 300000 parameters and a of simple sequential (where each layer is connected to a layer before and after) structure. As an example, a well known model VGG16

[28] has 138 million parameters. Although depending on a specific task it is quite common to measure the number of parameters in millions.

Model training
Having relatively small model and 3.5 million images one iteration of computations on the whole data set takes up about 50 hours. A number of epochs is determined in the process of training by following loss and accuracy curves, however practically it may vary from tens to hundreds. Having the least number, say 10 epochs our process of training would last for 500 hours which is about 20 days. Therefore we used pre-selected representative data sets for various models.

On the average training of autoencoder lasted about 6 hours and about 10 for a classifier (experimenting with representative set selection is taken into account).

Saving bottleneck features During training process we could use a smaller pre-selected data set this way reducing the time resources. However, we also need to compute the bottleneck vectors for each of 1048 whole images. This lasts about 10-12 hours.

Another very important aspect is that we did not need any image labeling since we trained

- autoencoders that do not need any information about the image at all
- classifier that was given a few labels of classes that are based on very common information about a patient, provided anyway.

This way lots of human resource hours are saved.

To sum up, completing a full size feature extraction process we would need multiple GPUs and months of trials. Our approach sought to check whether we can get meaningful results by strongly scaling down the resources. Also our modeling process does not need additional labeling which would mean quite expensive resource - time of a specialist.

3.8 Visualizations

In this chapter we suggest one possible approach to visually review the results - to explore patches with maximal values of the significant bottleneck variables. These are the steps:

1. First, we set a threshold value Tr_i for a particular bottleneck covariate B_i. In our case we compared two cases: $Tr_i = 0$ and bottleneck's 0.95 quantile $Tr_i = B_{(n*0.95)}$ throughout the whole data base patches (including all the WSI). Note, that the efficient ways of choosing the Tr_i values are left for further investigation.
2. Second we show the whole image with marked patches where $B_i \geq Tr_i$.
3. For each B_i we may chose different coloring.

Next we illustrate the visualization approach.

1. Two bottleneck values of significance from Classifier CNN was: cm_123 and cm_390. We find the threshold values for each of them: thresh_123 = 3.06 and thresh_390 = 2.45. Another threshold for both B_i as mentioned are 0. The distributions of both bottlenecks in all the patches are given in Figure 11. (Maximal values are 10.59 and 9.01 correspondingly). Note, that negative bottleneck values go under 0 value in the histogram, because ReLU function is applied to bottleneck layer.

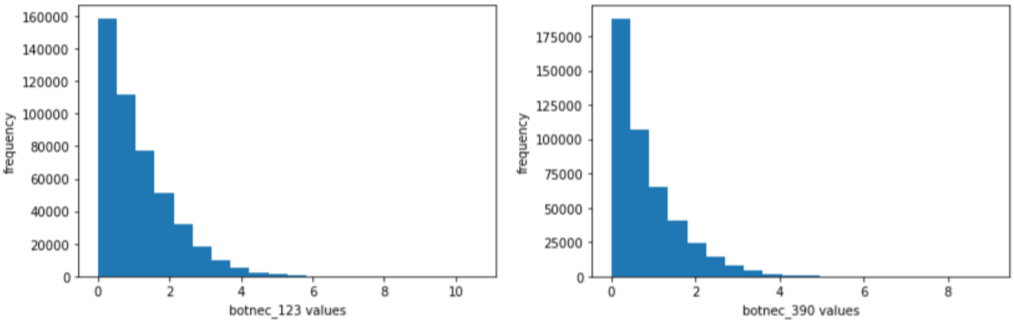

Figure 11: Distribution of cm_123 and cm_390 values

2. The illustrative visualization of patches which corresponding to activated informative significant covariates are given in Figure 12. On the left there is a view having a computed threshold $Tr_i = B_{(n*0.95)}$, on the right $Tr_i = 0$. Blue color is for cm_390, red for cm_123.

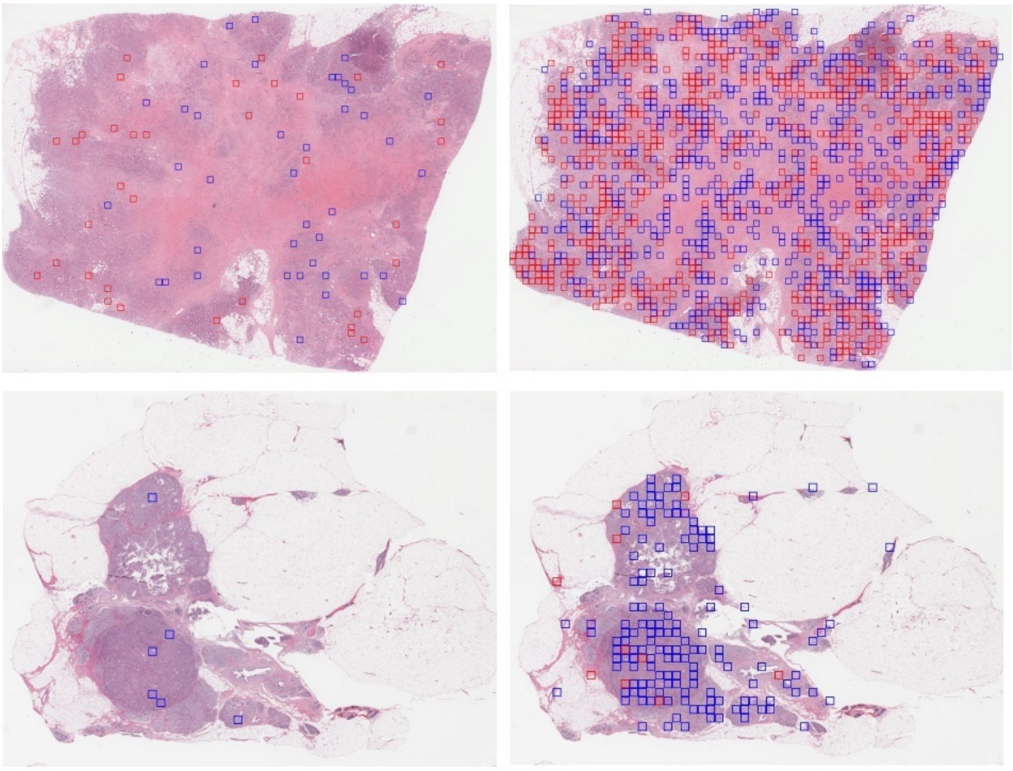

Figure 12: Image patches which activating the informative covariates.

4 Conclusions

We analyzed The Cancer Genome Atlas Program (TCGA) data including diagnostic images and general case specific information variables. Firstly, the base data set was produced - including only time, censoring, fixed significant covariates. Then by running Cox proportional hazards regression - base model produced.

Secondly, we collected additional tissue image data, then we trained two groups of convolutional neural network models: autoencoders and classifiers - diagnostic image patches were used for such analysis. We compared 3 models: narrow-bottleneck model, one best from autoencoders, one best from classifiers. The bottleneck representations (vectors of feature-wise maximum / average values along all the patches from the same whole image) were calculated for each diagnostic slide. Excepting Cox regression analysis was run on two data sets:

- bottleneck-only - table containing only time, censoring and informative variables (bottleneck values) received from tissue images
- bottleneck-full - table containing time, censoring, basis fixed covariates and informative variables (bottleneck values) received from tissue images

The two models per autoencoder / classifier were produced. Finally we compare each of bottleneck model with basemodel. The results are given in the table.

Model	AIC	Concordance	log-likelihood
Basemodel	1270	0.73	72.04 on 6 df
Autoencoder bottleneck-only model (max)	1494	0.63	25.32 (7 df)
Autoencoder bottleneck-full model (max)	-	-	-
Classifier bottleneck-only model (max)	1484	0.66	31.18 (5 df)
Classifier bottleneck-full model (max)	1247	0.78	98.87 (8 df)
Narrow-bottleneck model	1579	0.66	54.53 (64 df)

In conclusion:

1. Proposed approach to use classification CNN allows to extract meaningful features for survival estimation in low-mid computational resource conditions.
2. Both proposed models based on convolutional neural networks produced significant new covariates (high level features), but only classification CNN features combined with the basis features can improve the Cox regression model.
3. Both autoencoder and classifier bottleneck-only models without fixed information did not outperform basis model, suggesting that extracted high level features are not informative enough to supersede fixed variables.
4. The selection of convolutional model for image feature extraction plays a significant role.
5. Deep feature vector high dimensionality problem is better solved by feature pre-selection the usage of poor model producing lower-dimensional output at once.

References

[1] A. Agarwalla, M. Shaban, and N. M. Rajpoot. Representation-aggregation networks for segmentation of multi-gigapixel histology images. *arXiv preprint arXiv:1707.08814*, 2017.

[2] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.

[3] D. Chen, S. Liu, P. Kingsbury, S. Sohn, C. B. Storlie, E. B. Habermann, J. M. Naessens, D. W. Larson, and H. Liu. Deep learning and alternative learning strategies for retrospective real-world clinical data. *NPJ digital medicine*, 2(1):1–5, 2019.

[4] J. Cong, B. Wei, Y. Yin, X. Xi, and Y. Zheng. Performance evaluation of simple linear iterative clustering algorithm on medical image processing. *Bio-medical materials and engineering*, 24(6):3231–3238, 2014.

[5] L. A. Cooper, E. G. Demicco, J. H. Saltz, R. T. Powell, A. Rao, and A. J. Lazar. Pancancer insights from the cancer genome atlas: the pathologist's perspective. *The Journal of pathology*, 244(5):512–524, 2018.

[6] A. Cruz-Roa, H. Gilmore, A. Basavanhally, M. Feldman, S. Ganesan, N. N. Shih, J. Tomaszewski, F. A. González, and A. Madabhushi. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific reports*, 7:46450, 2017.

[7] L. Cui, H. Li, W. Hui, S. Chen, L. Yang, Y. Kang, Q. Bo, and J. Feng. A deep learning-based framework for lung cancer survival analysis with biomarker interpretation. *BMC bioinformatics*, 21(1):1–14, 2020.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] I. Hadji and R. P. Wildes. What do we understand about convolutional networks? *arXiv preprint arXiv:1803.08834*, 2018.

[10] M. Häfner, M. Liedlgruber, A. Uhl, A. Vécsei, and F. Wrba. Color treatment in endoscopic image classification using multi-scale local color vector patterns. *Medical image analysis*, 16(1):75–86, 2012.

[11] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern recognition*, 42(3):425–436, 2009.

[12] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 2424–2433, 2016.

[13] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[14] T. Iwatsubo. Alzheimer's disease neuroimaging initiative (adni). *Nihon rinsho. Japanese journal of clinical medicine*, 69:570, 2011.

[15] A. Janowczyk and A. Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of pathology informatics*, 7, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[17] Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[18] Y. Li and W. Ping. Cancer metastasis detection with neural conditional random field. *arXiv preprint arXiv:1806.07064*, 2018.

[19] D. Liu, Y. Liu, S. Li, W. Li, and L. Wang. Fusion of handcrafted and deep features for medical image classification. In *Journal of Physics: Conference Series*, volume 1345, page 022052. IOP Publishing, 2019.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[21] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner. Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, 19(9):1498–1507, 2007.

[22] D. Nie, H. Zhang, E. Adeli, L. Liu, and D. Shen. 3d deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In *International conference on medical image computing and computer-assisted intervention*, pages 212–220. Springer, 2016.

[23] P. J. Phillips, A. N. Yates, Y. Hu, C. A. Hahn, E. Noyes, K. Jackson, J. G. Cavazos, G. Jeckeln, R. Ranjan, S. Sankaranarayanan, et al. Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. *Proceedings of the National Academy of Sciences*, 115(24):6171–6176, 2018.

[24] S. Pölsterl, I. Sarasua, B. Gutiérrez-Becker, and C. Wachinger. A wide and deep neural network for survival analysis from anatomical shape and tabular clinical data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 453–464. Springer, 2019.

[25] T. Qaiser and N. M. Rajpoot. Learning where to see: A novel attention model for automated immunohistochemical scoring. *IEEE transactions on medical imaging*, 38(11):2620–2631, 2019.

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

[27] H. Shih-Chung, I.-C. Chang, and H. Chung-Lin. Object verification in two views using sparse representation. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 504–509. IEEE, 2016.

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] R. Szeliski. *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.

[30] H. R. Tizhoosh and L. Pantanowitz. Artificial intelligence and digital pathology: challenges and opportunities. *Journal of pathology informatics*, 9, 2018.

[31] R. van Domburg, S. Hoeks, I. Kardys, M. Lenzen, and E. Boersma. Tools and techniques–statistics: how many variables are allowed in the logistic and cox regression models. *EuroIntervention*, 9(12):1472–1473, 2014.

[32] M. Veta, Y. J. Heng, N. Stathonikos, B. E. Bejnordi, F. Beca, T. Wollmann, K. Rohr, M. A. Shah, D. Wang, M. Rousson, et al. Predicting breast tumor proliferation from whole-slide images: the tupac16 challenge. *Medical image analysis*, 54:111–121, 2019.

[33] H. Wang, A. C. Roa, A. N. Basavanhally, H. L. Gilmore, N. Shih, M. Feldman, J. Tomaszewski, F. Gonzalez, and A. Madabhushi. Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. *Journal of Medical Imaging*, 1(3):034003, 2014.

[34] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[35] Y. Wu and Q. Ji. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127(2):115–142, 2019.

[36] E. Wulczyn, D. F. Steiner, Z. Xu, A. Sadhwani, H. Wang, I. Flament-Auvigne, C. H. Mermel, P.-H. C. Chen, Y. Liu, and M. C. Stumpe. Deep learning-based survival prediction for multiple cancer types using histopathology images. *PLoS One*, 15(6):e0233678, 2020.

[37] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206, 2007.

[38] W. Zhu, L. Xie, J. Han, and X. Guo. The application of deep learning in cancer prognosis prediction. *Cancers*, 12(3):603, 2020.