

THE UNIVERSITY OF LIVERPOOL

PHD THESIS

**The Development of Machine
Learning Methods for Head and
Neck Cancer Prognosis**

Author:

Conor Whitley

Supervisors:

Dr. David Martin

Dr. Steve Barrett

Professor Marta Garcia-Finana

Dr Ruwanthi

Kolamunnage-Dona

Abstract

Contents

Abstract	iii
List of Figures	vii
List of Tables	ix
1 Deep Learning Prognostic Tools	1
1.1 Introduction	1
1.2 Materials and Methods	2
1.2.1 Optimisation of network structure	7
1.3 Results	10
1.4 Discussion	13
1.5 Conclusion	17

List of Figures

1.1	A 2D convolutional layer	3
1.2	An typical convolutional network example pipeline; parameters associated with each step are shown in green; extra parameters are shown in bold.	5
1.3	A simplified schematic of the optimal one-dimensional CNN architecture. The shape of the data as it passes through each layer is represented by vectors; the colour of each vector represents a different kernel. Intermediate layer activations, regularisation steps etc are represented by orange boxes. The final element represents the probability of a poor prognosis for that spectrum.	8
1.4	Multilayer perceptron neural network	10
1.5	Median ROC and PR curves shown in solid lines; dashed lines represent baselines scores associated with random chance. AUROC and AUPRC scores are shown for each set of prognostic indicators.	11
1.6	Whisker boxplots of classification statistics calculated across all data subsets. AUROC (A); AUPRC (B); F1 (C); MCC (D); specificity (E); sensitivity (F); PPV (G); NPV (H). Boxes show the median, 25 th , and 75 th percentiles; whiskers extend to points that lie within 1.5 inter quartile ranges of the lower and upper quartiles; points lying outside this range are shown as individual diamonds.	12
1.7	Kaplan-Meier survival curves of predicted risk groups	13

List of Tables

1.1	Optimal convolutional neural network parameters.	8
1.2	Optimal multilayer perceptron network parameters.	9
1.3	Median classification statistics	13

1 Deep Learning Prognostic Tools

1.1 Introduction

In recent years research a subcategory of machine learning called 'deep learning' has rapidly emerged to become the state-of-the-art in field of artificial intelligence. Deep learning is able to achieve superior performance where previous approaches have fallen short, and has facilitated the development of many useful applications [1]. Deep learning is a broad term covering a large swathe of mathematical models all comprising a multi-layered ANN of some form. The recent surge in interest in ANNs has been driven by a number of factors. In the last twenty years computing power, storage, and the availability of data have increased dramatically [2]. This has facilitated a paradigm shift in the way that artificial intelligence systems are created; modern algorithms have been developed which learn from data very efficiently, rather than being explicitly programmed by a human to accomplish a given task. Due to the highly parametric nature of deep ANNs, large quantities of data are required to obtain optimal model configurations, this data-focused approach is responsible in large part for the recent success of ANNs [2].

To evaluate the potential of CNNs as a prognostic model for oral cancer, a one-dimensional CNN was created to establish if any improvement was gained by adding spectral information. The network follows a similar structure to a 2D CNN; the objective is that the network would extract higher level features from

raw absorbance values — potentially correlating to levels of known biochemicals. A crucial factor in the transmission of FTIR microscopy into a clinical setting is its ability as a technique to be universally adopted. To do this factors including sample preparation, measurement environment, and measurement technique must be as uniform as possible. Preprocessing data is an attempt to mitigate the effects of potential inconsistencies in measurement practice but the process is never perfect. It would be desirable for any analysis method to be able to obviate the need for preprocessing all together by being robust to invariances introduced by experimental practice alone. A CNN is able to manage this to some extent due to it expressing translational invariance — attributable to the the convolution [3], and pooling layers [4, 5]. By effectively scanning the entire spectrum and learning how to recognise patterns spanning multiple wavenumbers, the model is robust to slight alterations in wavelength-dependent absorbance.

1.2 Materials and Methods

Convolutional neural networks

As outlined in ?? an ANN consists of a number of layers, each of which comprise a number of nodes. Nodes in a NN are a representation of a relatively simple equation known as the perceptron equation (??); which are combined in complex ways and used as a highly parametric model of a particular inference problem. A neural network trained using labelled data will optimise free parameters within the network to minimise a loss function constructed for the problem at hand. In particular, neural networks have allowed for advances in applications where data contains temporal or spatial information. Convolutional neural networks (CNNs) are a type of network containing specialised layers capable of extracting spatial information from data. This is accomplished with the

use of a kernel which is convolved over the input data. A kernel comprises a number of parameters which are refined during a training phase to capture the most relevant spatial information. The first few layers of a CNN are utilised as a method of feature extraction; values of the convolution kernels are refined progressively to extract useful spatial features in the data; these features are then fed into a standard MLP network for further feature extraction, and later classification or regression.

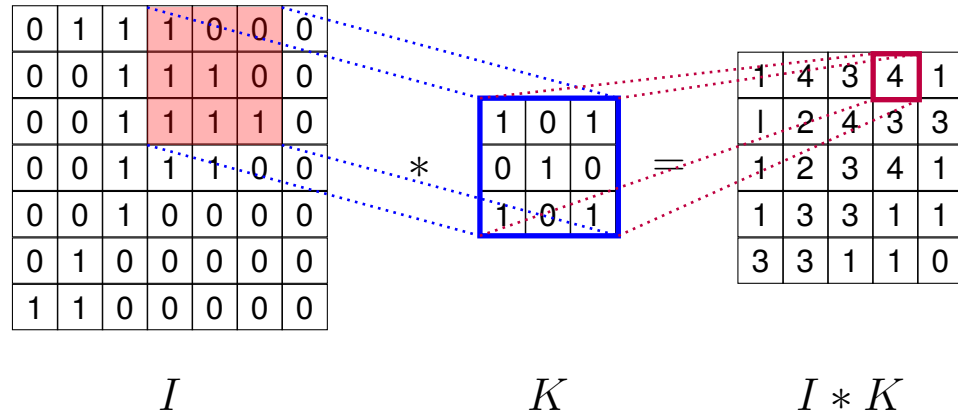


FIGURE 1.1: A 2D convolution layer showing a simple (3×3) kernel K convolved over an input image I of size (m×n). The resulting convolution $I * K$ is effectively a spatial map of where in I most closely resembles K .

This convolution operation can be expressed formally as:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (1.1)$$

The kernel K is moved across the dimensions of the data where it is multiplied by the values in the kernel. A mapping of the similarity of the data to the kernel at that point is obtained then aggregated, typically a maximum or mean is taken and the values are pooled and assigned to an output. The behaviour of a kernel layer is dictated by its size, stride — the number of elements the kernel shifts per iteration, and dilation — the mapping of kernel elements to non contiguous elements of the data. Like other layer types convolutional layers can be

altered by many parameters but will not be discussed further here. Depending on the dimensionality of the data the kernel can be combined with other kernels to find correlations between them; allowing colour information in 2D images to be used. A bias term is added to the pooled value and subjected to an activation function like in a normal perceptron layer. The next layer is obtained by convolving over the preceding layer using another kernel, this continues for a number of layers. The input is flattened into a one-dimensional vector where it is passed into a normal MLP where it reaches a softmax layer output. This softmax layer eq. (1.2) outputs scores which sum to one and can be loosely interpreted as probabilistic predictions of a given class.

$$\text{softmax}_k(x) = \frac{e^{W_k^T x}}{\sum_{i=1}^n e^{W_i^T x}} \quad (1.2)$$

Intermediate steps between these layers can be introduced to assist with regularisation such as dropout layers, batch normalisation layers [6], and many others. Dropout layers [7] are often included in ANN architectures as they provide a strong regularisation effect during the training phase. A simplified representation of a CNN showing the sequential nature of the described layers is shown in fig. 1.2. Dropout layers are typically implemented in a similar way to regular hidden layers, however they differ in that when any forward pass occurs in the training stage a node may become inactive, preventing any change in the weights of connected nodes for that particular pass. Each dropout layer typically has a probability associated with it which dictates the chance of becoming inactive. Dropout works effectively by discouraging weights from converging towards similar values — encouraging redundancy in the network structure.

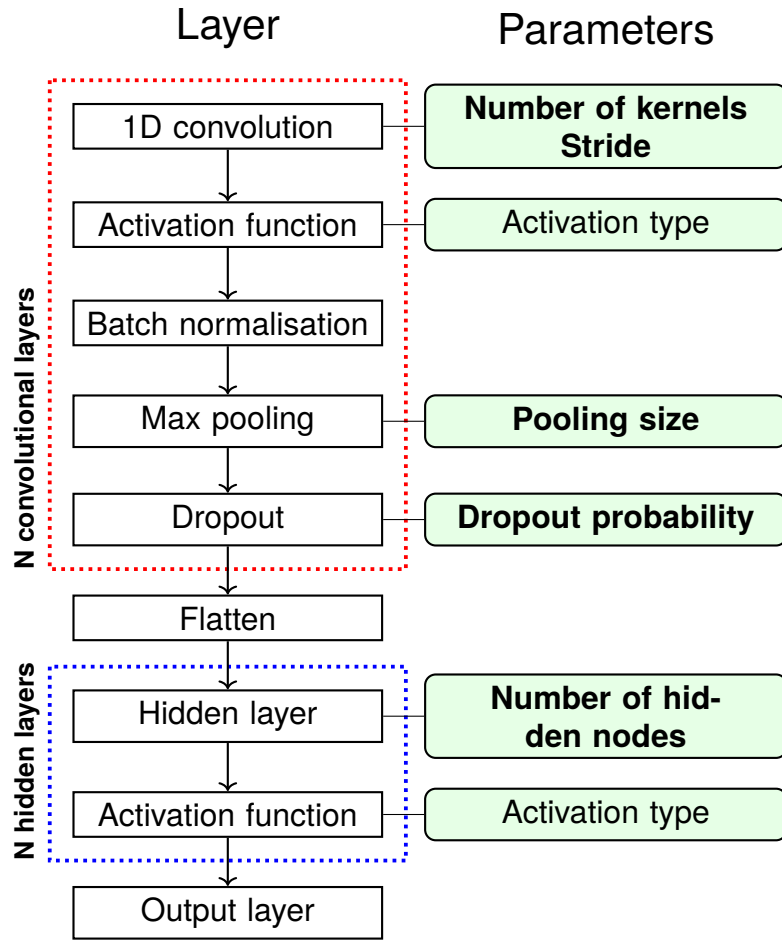


FIGURE 1.2: An typical convolutional network example pipeline; parameters associated with each step are shown in green; extra parameters are shown in bold.

Batch normalisation layers were included in the CNN model to increase the speed and efficiency of the training procedure. Batch normalisation layers work by normalising the distribution of values flowing from one set of nodes to the next, by scaling to a given range. This helps to prevent issues associated with exploding/vanishing gradients whereby update values for nodes increase or decrease rapidly to the detriment of the training process.

Furthermore, many parameters are associated with aspects of the training procedure of the network itself. The objective of the optimisation procedure used to train neural networks is to maximise or minimise a score with respect to the network parameters. Number of optimisation techniques are used but stochastic gradient descent (SGD) was the chosen method for both ANNs described

here. SGD works by computing this objective score on a randomly selected subset of the training data. This is beneficial for large datasets where computing on the entire set may be computationally infeasible. When an update occurs a coefficient called the *learning rate* is used to dictate the weighting of the new value. Another parameter used to influence the optimisation strategy is the *weight decay*; a coefficient used to alter the effect of the gradient value on the objective function. A thorough description of SGD and its associated parameters is available from [8]. Another commonly used technique to improve convergence during the training stage is to initialise neural network weights. Weights were initialised using the method described by Kaiming [9]; initialising layer weights has been shown to decrease convergence times and improve the stability of the optimisation procedure.

To evaluate the benefit of utilising convolution layers to analyse spectra, a comparison of a CNN model was made with a MLP. Both ANN models were constructed, trained, and evaluated using an open-source python library PyTorch [10]. Additional packages [11, 12, 13] were leveraged to implement the evaluation procedure along side other common machine learning operations. The same procedure as discussed in ?? was followed; out of bag sampling was utilised to obtain distributions of classification values, inverse weighting was used to mitigate the effects of dataset imbalance.

The dataset comprised FTIR spectra taken from primary tumour sites of 29 patients with a diagnosis of OSCC. Inclusion criteria for this study were as previously described in ??: a diagnosis of OSCC; the presence of OSCC in the TMA core; the ability to co-register adjacent H&E stained and FTIR imaged sections; a follow-up period after surgery of at least 24 months; HPV negative. Images were acquired at a resolution of 6cm^{-1} over a spectral range of 990cm^{-1} to 3800cm^{-1} using a co-addition of 128 scans. Attenuator and integration time of the focal plane array (FPA) were chosen to gain the maximum signal-to-noise

ratio. Background scans were acquired using a blank CaF_2 disk situated within the perspex box before each session of measurements.

The preprocessing steps required for each type of network are slightly different. Given that the convolutional layers in the CNN model are used to extract features from multiple wavenumbers simultaneously, the only preprocessing step is to normalise the data. For the MLP model vector normalisation was used in order to account for sample thickness; wavenumber absorbance features were mean-centered; and variance scaled to one; before a final PCA step to reduce dimensionality of the dataset.

1.2.1 Optimisation of network structure

The sheer number of tunable parameters associated with ANNs necessitates a hyperparameter search similar to that described in ???. The open source optimisation framework Optuna [14] was used to determine an optimal network structure and associated hyperparameters. The hyperparameters in bold in ??? were chosen for optimisation for the CNN network; in addition the learning rate and weight decay parameter were included as optimal values are task-dependent and have a large impact on the training efficiency of ANNs [15, 16]. The median AUROC value was calculated across a five-fold cross validation of data subsets to determine the general suitability of the network configuration. Fifty sequential trials were chosen to allow sufficient exploration of the parameter space.

Convolutional network

A summary of the configuration determined by the procedure is given in table 1.1

TABLE 1.1: Optimal convolutional neural network hyperparameters and values.

Parameter name	Value
N convolutional layers	5
N kernels in convolution layer 1	96
N kernels in convolution layer 2	128
N kernels in convolution layer 3	32
N kernels in convolution layer 4	80
N kernels in convolution layer 5	128
Maxpool 1 size	3
Maxpool 2 size	3
Maxpool 3 size	7
Maxpool 4 size	5
Maxpool 5 size	3
N fully connected nodes	80
Dropout probability	0.45
Learning rate	8×10^{-5}
Optimum value	0.84

A simplified diagram of the CNN network ~~fig. 1.3~~ configuration determined by the optimisation procedure is shown ~~below~~. The CNN network contains five convolutional and maxpooling layers of varying sizes. Deeper network designs are typically better at extracting high-level structural information in data [17].

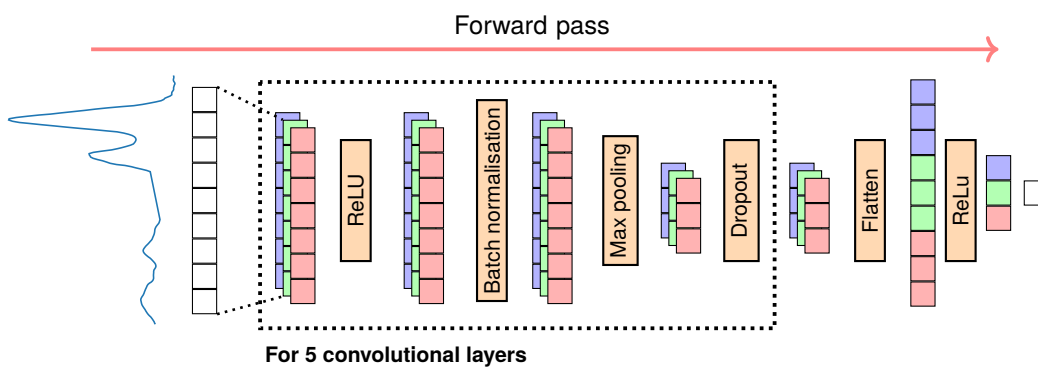


FIGURE 1.3: A simplified schematic of the optimal one-dimensional CNN architecture. The shape of the data as it passes through each layer is represented by vectors; the colour of each vector represents a different kernel. Intermediate layer activations, regularisation steps etc are represented by orange boxes. The final element represents the probability of a poor prognosis for that spectrum.

Multilayer perceptron network

A summary of the optimal network configuration for the MLP determined by the optimisation procedure is given in table 1.2. A simplified representation of the optimal MLP network is shown in fig. 1.4. The configuration is a relatively shallow MLP with two layers. Both layers have a strong regularisation effect applied by dropout layers with dropout probabilities ~ 0.3 . The first layer has 169 nodes followed by ten in the second layer; a potential explanation for this is that there are fewer higher level features needed in the second layer to achieve a good level of discrimination between risk groups.

TABLE 1.2: Optimal MLP network parameters.

Parameter name	Value
N hidden layers	2
N nodes in hidden layer 1	169
N nodes in hidden layer 2	10
Dropout layer 1 probability	0.28
Dropout layer 2 probability	0.29
Learning rate	2×10^{-5}
Weight decay	1.1×10^{-3}
Optimum value	0.77

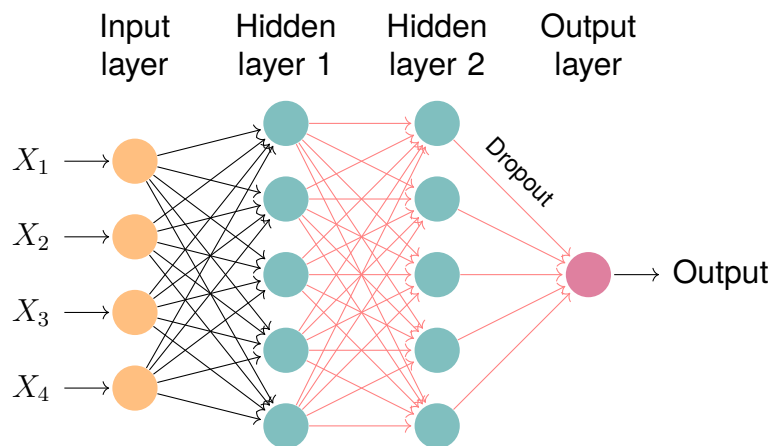



FIGURE 1.4: A multilayer perceptron neural network with an input layer consisting of four input variables $x_0 \dots x_4$, three hidden layers of five nodes each, and a single output layer. Dropout layers are represented by red arrows between network layers.

1.3 Results

As discussed in detail in ??, the objective was to predict which risk group a patient falls into, risk groups were determined by a GA optimisation routine seeking to achieve the maximum prognostic information. Predictions of risk groups for each patient were taken as the median probability predicted across all spectra for any given patient. The threshold used to dichotomise probabilistic predictions was set by maximising the MCC score. The MCC score considers all possible prediction outcomes and is a well-rounded measure of performance for discrete predictions. 

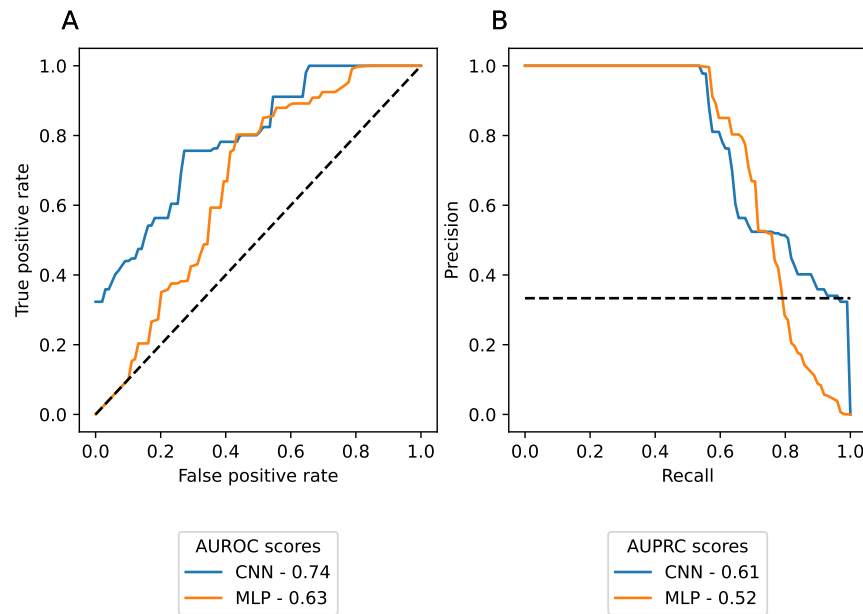


FIGURE 1.5: Median ROC and PR curves shown in solid lines; dashed lines represent baseline scores associated with random chance. AUROC and AUPRC scores are shown for each set of prognostic indicators.

Median ROC and PR curves [fig. 1.5](#) indicate that both models show some utility as classifiers. The AUPRC scores for both classifiers are both significantly above the baseline score — Indicating that both models can balance both statistics effectively, and that imbalance in the dataset was not detrimental (Figure 1.5[B]). The AUROC score is modest for the MLP model at 0.63, the CNN model performs better across all classification thresholds with a score of 0.74.

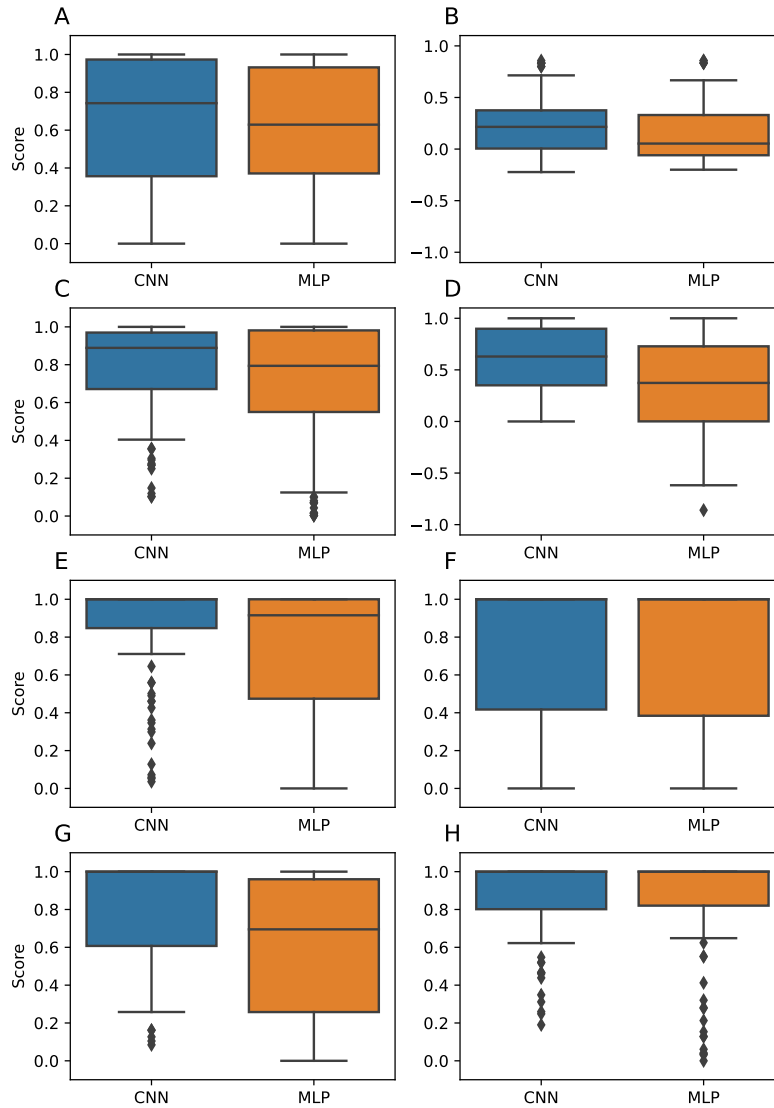


FIGURE 1.6: Whisker boxplots of classification statistics calculated across all data subsets. AUROC (A); AUPRC (B); F1 (C); MCC (D); specificity (E); sensitivity (F); PPV (G); NPV (H). Boxes show the median, 25th, and 75th percentiles; whiskers extend to points that lie within 1.5 inter quartile ranges of the lower and upper quartiles; points lying outside this range are shown as individual diamonds.

Score distributions fig. 1.6[A-H] for both models are generally very strong. The CNN model is superior for all statistics and generally varies less than the MLP model despite the low threshold value. The low threshold value for the CNN model indicates that the distribution of prediction scores spans a low range.

Kaplan-Meier curves were plotted for groups dichotomised by the threshold in

TABLE 1.3: Median classification statistics. Classification thresholds (table 1.3) used to dichotomise prediction probabilities were determined to be those that maximised the MCC score.

Variables	AUROC	AUPRC	F1	MCC	Spec	Sens	PPV	NPV	Thresh
CNN	0.74	0.22	0.89	0.63	1.00	1.0	1.0	1.0	0.01
MLP	0.63	0.05	0.79	0.37	0.92	1.0	0.7	1.0	0.58

table 1.3. Figure 1.7[A] shows a clear distinction between groups determined by the CNN classifier; a significant p-value was obtained by performing a log-rank test using the predicted groups. Figure 1.7[B] shows effectively no ability to discriminate between risk groups, which is in agreement with an insignificant p-value of 0.4.

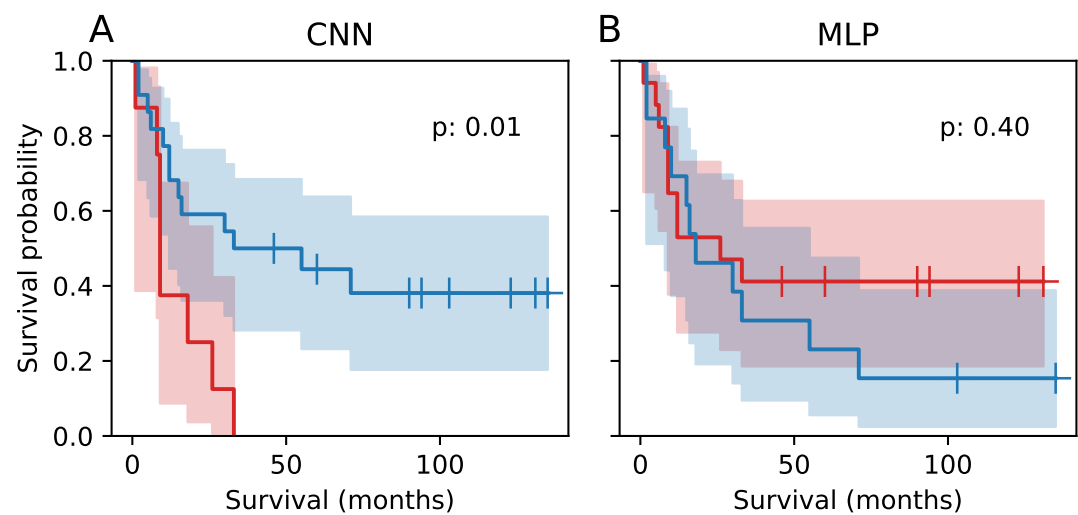


FIGURE 1.7: Kaplan-Meier survival curves of predicted risk groups for CNN [A] and MLP [B] classifiers.



1.4 Discussion

Deep learning methods are a popular choice for a variety of classification tasks in the field of medical diagnostics, due to increasing availability of data and computing resources. Deep learning methods coupled with vibrational spectroscopy data sets are also rapidly finding more use in cancer diagnostics [18].


The development of CNNs has enabled researchers to utilise structural information present in data to quantify spatially and temporally dependent features. FTIR spectra possess such structural information due to the overlap of absorbance bands associated with various chemical moieties present in a sample. The use of multiple convolution layers facilitates the extraction of high-level structural information present in data, determining patterns from multiple wavenumbers simultaneously in a similar way to what might be done by a human searching for known peaks in chemical spectra. Even slight shifts in absorption values induced by inconsistent measurement technique could result in the miss-classification of a spectrum, negatively affecting the prospect of FTIR being widely adopted as a diagnostic tool.

The work described here is an attempt to establish the viability of CNNs coupled with FTIR spectroscopy as prognostic prediction tools. The objective is to correctly predict the prognosis of a patient from spectra measured from primary tumour sections taken from a TMA array. An initial optimisation procedure was performed to determine a suitable network architecture; the Bayesian optimisation procedure explored a large parameter space seeking to maximise the median AUROC of a five-fold cross validation routine. Using the optimal network configurations, both ANNs were evaluated using a sampling without replacement bootstrap strategy to obtain distributions of classification scores.


ROC and PR analysis ~~fig. 1.5[A,B]~~ was used to estimate the general utility of each ANN configuration. AUROC scores of 0.74 and 0.63 for the CNN and MLP networks respectively show that both classifiers have some utility across all classification thresholds. AUPRC scores of 0.61 (CNN) and 0.52 (MLP) show that both models display some utility despite some imbalance in the dataset. Classification statistics ~~Figure 1.6[A-G]~~ are generally very strong for both models; a median MCC score of 0.63 for the CNN model indicates a very strong classifier, with the MLP achieving a median score of 0.37.


Survival curves for groups predicted by the CNN model ~~fig. 1.7[A]~~ show good separation between risk groups; a significant p-value of 0.01 given by a log-rank suggests this is in agreement  Interestingly despite the good overall classification performance of the MLP model, the threshold which maximised the classification scores resulted in poor prognostic predictions. 





Much of the existing literature concerning prognostic indicators utilises disease specific survival and overall survival as indicators of patient prognoses. Stratification of these measures into groups according to risk is typically on a set number of years e.g. one year, two years. The decision to use a cut off threshold of a discrete number of years is somewhat arbitrary, thus, it would be desirable to determine a threshold that was decided objectively. A GA approach discussed in ?? was used to stratify patients into either a low or high risk group. This threshold was determined to be 11 months and was used to dichotomise patients into risk groups which served as prediction objectives for the CNN and MLP models.

MLPs have ~~already~~ been utilised with vibrational spectroscopy methods to detect breast cancer in ATR-FTIR data [19]. The authors used three ANNs in a 10-fold cross validation scheme to discriminate between FTIR spectra collected from 78 malignant and 88 benign breast tumours. ~~The authors~~ found that ANNs had superior performance across many classification statistics in comparison to many other classifiers. The ANN models however did not perform as well as a SVM classifier on the same data. 

Existing work [20] has leveraged 2D CNNs to discriminate between different grades of breast cancer from a cohort of 96 patients. The study followed a similar analysis routine to that discussed in ??, where a series of subsets of the data were drawn without replacement to obtain distributions of classification scores. It was found that the addition of spatial information from convolutional layers improved the model performance considerably over pixel-level

predictions of a large range of models including: SVM, random forest, and an RBF kernel. Adding spatial information increased the overall accuracy of the predictions by $\sim 20\%$; the specificity and sensitivity increased considerably with one class improving by $\sim 60\%$. 

A one-dimensional neural network was used to classify FTIR, Raman, and near infrared (NIR) data derived from a variety of food samples [3]. The ANN developed by the authors was a shallow CNN,  which improved on overall accuracy scores of other classifier methods on preprocessed data from 62% to 86%; and from 89% to 96% on raw data. CNNs were used with success on ATR-FTIR data in forensics to detect synthetic cannabinoids [21]. The authors found that CNNs were capable of identify synthetic cannabinoids, achieving 98.7% accuracy and an F1 score of 98.5% — meeting the standards of a forensic screening system. CNNs applied to vibrational spectroscopy have enjoyed further success in forensic applications where they employed to identify amphetamines with an accuracy of over 90% [22].

 A rigorous analysis procedure was performed to obtain distributions of classification scores, and to gain an insight into the general applicability of each model to unseen data. The relatively small sample set was a key issue facing this study due to the expense of acquiring and imaging large numbers of samples.  large degree of variation was observed across some classification statistics, potentially indicating a large degree of biological heterogeneity in the dataset — a known characteristic of OSCC primary tumours [23, 24]. A potential cause for this could be the effect of inherent molecular heterogeneity of the tumour microenvironment [25]; or perhaps varying extents of lymphocyte infiltration present in specimens. The difficulty of annotating samples is also likely to introduce noise into the dataset; alongside inconsistencies in sample preparation and measurement procedure  

1.5 Conclusion

For FTIR spectroscopy to transition into widespread clinical use as a prognostic tool, measurement errors must be mitigated wherever possible. CNNs have the potential to mitigate some of these effects due to the usage of convolution operations as a means of extracting useful features from FTIR spectra. This work has shown that through the use of an optimisation procedure it was possible to use CNNs to correctly classify FTIR spectra derived from primary tumour sites into useful risk groups. The CNN model evaluated here showed superior classification performance over a comparable MLP network architecture when evaluated using a number of conventional metrics. A thorough out of bag bootstrap procedure was used to obtain distributions of classification scores to estimate the variability of these scores. The usage of these models could facilitate the ethical selection of patients for neo-adjuvant treatment in clinical window-trials whilst minimising overtreatment. This could be a crucial first step to improving the range of treatment options available to patients with OSCC.

Bibliography

- [1] Grigorios Tsagkatakis, Anastasia Aidini, Konstantina Fotiadou, Michalis Giannopoulos, Anastasia Pentari, and Panagiotis Tsakalides. Survey of deep-learning approaches for remote sensing observation enhancement. *Sensors (Switzerland)*, 19(18):1–39, 2019.
- [2] Michael Haenlein and Andreas Kaplan. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, 61(4):5–14, 2019.
- [3] Jacopo Acquarelli, Twan van Laarhoven, Jan Gerretzen, Thanh N. Tran, Lutgarde M.C. Buydens, and Elena Marchiori. Convolutional neural networks for vibrational spectroscopic data analysis. *Analytica Chimica Acta*, 954:22–31, 2017.
- [4] Eric Kauderer-abrams. Quantifying Translation-Invariance in Convolutional Neural Networks.
- [5] Osman Semih Kayhan and Jan C Van Gemert. On Translation Invariance in CNNs : Convolutional Layers can Exploit Absolute Spatial Location. (class 2):14274–14285.
- [6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.
- [7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.
- [8] George E. Dahl, Tara N. Sainath, and Geoffrey E. Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout.

ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, (2010):8609–8613, 2013.

- [9] Kaiming He. Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification.
- [10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [12] Cameron Davidson-Pilon. lifelines: survival analysis in python. *Journal of Open Source Software*, 4(40):1317, 2019.
- [13] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and

- SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [14] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902, 2019.
- [15] Yanzhao Wu, Ling Liu, Juhyun Bae, Ka Ho Chow, Arun Iyengar, Calton Pu, Wenqi Wei, Lei Yu, and Qi Zhang. Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks. *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, pages 1971–1980, 2019.
- [16] Thomas M. Breuel. The effects of hyperparameters on SGD training of neural networks. *CoRR*, abs/1508.02788, 2015.
- [17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
- [18] Rasheed Omobolaji Alabi, Omar Youssef, Matti Pirinen, Mohammed El-musrati, Antti A. Mäkitie, Ilmo Leivo, and Alhadi Almangush. Machine learning in oral squamous cell carcinoma: Current status, clinical concerns and prospects for future—a systematic review. *Artificial Intelligence in Medicine*, 115:102060, 2021.
- [19] Rock Christian Tomas, Anthony Jay Sayat, Andrea Nicole Atienza, Jan-nah Lianne Danganan, Ma Rollene Ramos, Allan Fellizar, Kin Notarte Israel, Lara Mae Angeles, Ruth Bangaoil, Abegail Santillan, and Pia Marie Albano. Detection of breast cancer by ATR-FTIR spectroscopy using artificial neural networks. *PLoS ONE*, 17(1 January):1–24, 2022.

- [20] Sebastian Berisha, Mahsa Lotfollahi, Jahandar Jahanipour, Ilker Gurcan, Michael Walsh, Rohit Bhargava, Hien Van Nguyen, and David Mayerich. Deep learning for FTIR histology: leveraging spatial and spectral features with convolutional neural networks. *Analyst*, 144(5):1642–1653, 2019.
- [21] Catalina Mercedes Burlacu, Steluta Gosav, Bianca Andreea Burlacu, and Mirela Praisler. Convolutional Neural Network Detecting Synthetic Cannabinoids. pages 24–27, 2021.
- [22] Catalin Negoita, Mirela Praisler, and Iulia-Florentina Darie. Automatic identification of hallucinogenic amphetamines based on their ATR-FTIR spectra processed with Convolutional Neural Networks. *MATEC Web of Conferences*, 342:05003, 2021.
- [23] Sang Ik Park, Jeffrey P. Guenette, Chong Hyun Suh, Glenn J. Hanna, Sae Rom Chung, Jung Hwan Baek, Jeong Hyun Lee, and Young Jun Choi. The diagnostic performance of CT and MRI for detecting extracranial extension in patients with head and neck squamous cell carcinoma: a systematic review and diagnostic meta-analysis. *European Radiology*, 31(4):2048–2061, 2021.
- [24] Elham Alsahafi, Katheryn Begg, Ivano Amelio, Nina Raulf, Philippe Lucarelli, Thomas Sauter, and Mahvash Tavassoli. Clinical update on head and neck cancer: molecular biology and ongoing challenges. *Cell Death and Disease*, 10(8), 2019.
- [25] Patrick K. Ha, Steven S. Chang, Chad A. Glazer, Joseph A. Califano, and David Sidransky. Molecular techniques and genetic alterations in head and neck cancer. *Oral Oncology*, 45(4):335–339, 2009. Oral Cancer Management. Pitfalls and Solutions.