Machine Learning in Cancer Treatment: A Computational Method to Compile the Most Practical Treatment

New Mexico
Supercomputing Challenge
Final Report
April 7, 2021

Team 8
Clovis High Early College

Team Members:

- **Alexis Brandsma**
- **Erynn Vetterly**
- **Tristen Pool**

Contents

Executive Summary

Machine learning has already presented large-scale effectiveness in diagnosing cancer. However, how the field can be utilized to treat cancer remains underdeveloped. A model has been created using publicly available data to determine aspects of a given patient's cancer treatment, such as if one should receive chemotherapy. This model is automatically adaptable to any dataset in which the clinical data is in the .csv format with imagery in the DICOM format and corresponding patient IDs between them. Three datasets were utilized for the development of the model. Two of these datasets contain data from head and neck squamous cell carcinoma patients, and one contains data from breast cancer patients. It consists of three submodels: Clinical-Only, Image-Clinical, and CNN (convolutional neural network). These allow the model to maintain its performance across a variety of unique demands. In addition, the model contains an array of notable features such as support for a multi-target regression in the Clinical-Only and Image-Clinical submodels and a novel encoding/decoding algorithm so that text values can be utilized within the model. 91% test accuracy (n=296) has been achieved by the CNN submodel when dictating if a patient should receive chemotherapy, and additional promising results have been achieved on numerous other cancer treatment variables. Access to non-publicly available data, which is often quantitatively and qualitatively superior, could further increase the model's applications. Since cancer treatment is irreversible and every patient is unique, it remains unclear whether the model is capable of outperforming the human oncologists that fabricated the datasets chosen in practical applications. However, the model is capable of at least a more economical alternative and, at worst, an estimated 30% decrease in real-world accuracy compared to human oncologists.

Introduction

About Cancer

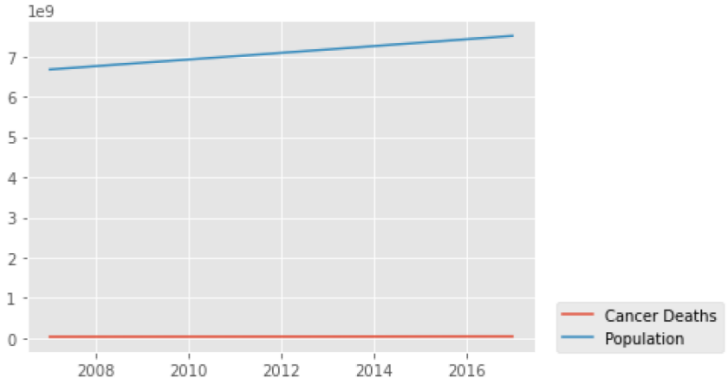

Figure 1 - Number of cancer deaths and population globally 2007-2017 [*Population, total*, n.d.; Ritchie & Roser, 2019]

Cancer is a disease caused by abnormal cells dividing uncontrollably. The cells do not die, and new cells are created unnecessarily. These excess cells form tumors [*What is Cancer?*, 2015]. Almost ten million people die from cancer annually [Roser & Ritchie, 2019]. Cancer has globally remained the second most common cause of death for more than 30 years. There are many types of cancer treatment. The process can be somewhat ambiguous as it is common for patients to receive a combination of treatments that can include surgery, chemotherapy, radiation therapy, immunotherapy, targeted therapy, or hormone therapy [*Cancer Treatment*, n.d.]. The type of treatment for a given patient is often chosen based on the type of cancer and how advanced it is. However, it can also be based on more broad factors such as the patient's general health and age.

Definition of Problem

Almost ten million people die from cancer annually [Roser & Ritchie, 2019]. Machine learning has presented effectiveness in diagnosing cancer. However, it remains less developed as to how the field can be utilized to treat cancer. Currently, compiling a cancer treatment plan can be a tedious process. A patient generally must consult numerous health professionals to receive the most desirable care [*How Treatment is Planned and Scheduled*, 2020]. This is not an ideal

practice as it results in unnecessary added costs and required resources. With millions within the United States unable to adequately access equitable and affordable health care overall, reducing care costs by increasing the presence of automation is ideal [Powell, 2016]. With the ambiguousness of cancer treatment planning and the quantity of data on the subject that is readily available, a machine learning model could be a feasible solution. A practical model for the previously described usage will be adaptable to any dataset and therefore usable for any cancer type. Although any target variable can be used in the model and testing will be done on several, a target variable that will be concentrated on is the variable in each dataset that indicates whether a patient has received chemotherapy or not. The delivery of this binary practice can vary by dataset, but it is commonly implemented with “yes, no”; or “1, 0.”

Datasets

Although virtually any CSV-DICOM dataset is compatible, three datasets were most significantly utilized to develop the computational model. The dataset referred to as HNSCC-HN1 is a set of head and neck squamous cell carcinoma patients (N=137) with clinical data and computed tomography (CT) [Wee & Dekker, 2019]. The dataset known as HNSCC contains imaging, radiation therapy, and clinical data from head and neck squamous cell carcinoma patients (N=627) at MD Anderson Cancer Center [Grossberg et al., 2020]. Lastly, the so-called METABRIC dataset contains clinical, m-RNA levels z-score, and gene mutations data for breast cancer patients (N=1904) [Alharbi, n.d.]. The non-imagery data for all of these are in the comma-separated values (CSV) format, while the imagery was in the Digital Imaging and Communications in Medicine (DICOM) format. DICOM is the standard format for the administration of medical imaging and its accompanying data [Charles, n.d.]. CSV is a widely utilized format for tabular data. Due to both formats being the standard for cancer data, they are what the model is compatible with. All datasets are publicly available.

Computational Method

In order to create a sensibly accommodating model, the computational model consists of three instances of machine learning components that function significantly differently from each other. This contributes to the objective of creating a model with the best performance, the least amount of hyperparameters, and the broadest range of dataset support. For instance, the ideal hyperparameters for a clinical-only dataset are unique from a clinical-image or image-only dataset. All except for the CNN component support a multi-target regression in which the number of outputs automatically adjusts based on how many target variables the user initially inputs.

Data Preprocessing

Preprocessing data is an essential aspect of the creation of the model. The datasets that have been gathered for use must be adapted into a more appropriate form for development. This adaptation includes removing missing data, reducing noise, and normalization [Miller, 2019]. It is also necessary to effectively partition the data into three categories: train, test, and validation. Validation is utilized to adjust the hyperparameters for the best performance and consists of 20% of the data chosen at random. After the hyperparameter tuning is finished, the test partition is used to evaluate the model's accuracy on new data. This partition is also 20% of the data. The train partition consists of 60% of the data. These three divisions are formulated after eliminating all rows with missing values. While this can often eliminate a significant portion from the original dataset, it is necessary to retain a valid dimension of the data and to avoid errors when creating the model due to NAN values. A common alternative to eliminating rows with missing values is to replace a given missing value with the mean of its column [Lee, 2019, 109]. However, this would not be practical because many datasets include text values, and the encoding algorithm is not suitable for this practice, especially for binary variables.

Figure 2 - How the contents of the METABRIC dataset are utilized

Encoding/Decoding Algorithm

A novel algorithm has been developed in order to utilize text within a dataset. It works by iterating through all characters in a given string and converting each character into its corresponding Unicode character code. Then the codes are appended into a single numerical value. This numerical value can be extremely large and can often cause issues with training the network even with normalization. To avoid this, the value is divided by ten times its length to retain its meaning yet not cause any issues. This always results in a float that is less than one and greater than 0.

Feature Selection

All machine learning components incorporate the same feature selection algorithm. This is to reduce noise in the data. Noise results from inapplicable features being present. It is essential to minimize noise for several reasons: generally better performance, decreased

overfitting, and less and more predictable computational expense [Shaikh, 2018]. The model is less deluded by only including features relevant to the target variable, and performance will likely increase. The algorithm works by measuring the correlation between the specified target variable and every other variable in the dataset. This correlation is defined by the Pearson Correlation Coefficient, which returns a value between -1 and 1. -1 indicates the strongest possible negative correlation between two variables. 1 indicates the strongest possible positive correlation. A value of 0 indicates no correlation. The model iterates through these values and selects the 10 variables that correspond with the highest absolute values. This leaves nine features excluding the target variable, which is guaranteed to have a value of 1 returned. Figure 3 depicts this target variable phenomenon. The distinctive yellow diagonal line in the visualization is due to the relation between two matching variables always having a fully positive correlation. When utilizing a multi-target regression, the same process is done but to each target variable individually. Then each set of nine features are used in the model. So for two target variables, 18 features would be used; for three target variables, 27 features would be used, and so on.

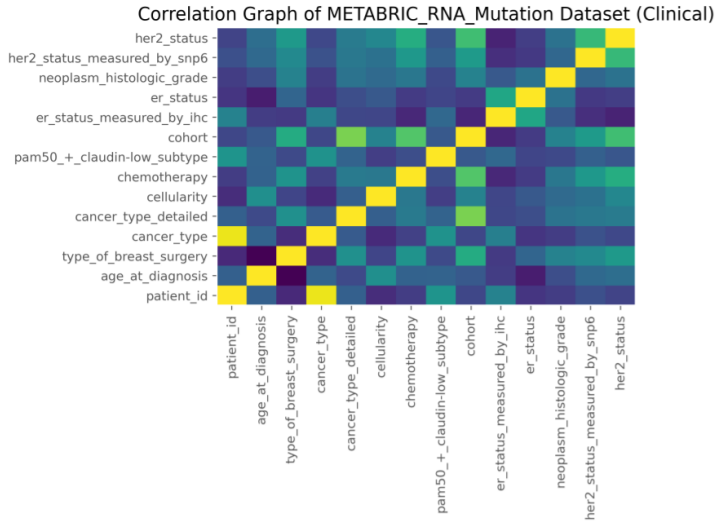

Figure 3 - Visualization of Pearson Correlation Coefficient for variables in the clinical element of the METABRIC dataset

Normalization

After feature selection and data partitioning, a normalization technique must be applied. The objective of normalization is to change the values of columns in a dataset to use a common scale without modifying the differences in the range of values. For instance, if the values of one

column are between 0 and 1 while the values of another column are between 300,000 to 400,000, the disproportion of these features could present issues when utilizing the model. The Sklearn *MinMaxScaler* object is used to execute this. It scales each feature individually to the range of 0 to 1. If there is more than one target variable, this process is also applied to the target data. Normalization is especially important in the model due to the nature of the encoding/decoding algorithm. It often outputs values that are precise and small and results in datasets that are very disproportionate from column to column.

Clinical-Only Model

The clinical-only submodel is designed to be utilized for the less computationally expensive clinical data (.csv) aspect of a given dataset only. The dynamics required for a model to perform acceptably on this type of data are significantly different from more computationally expensive data such as imagery. The clinical-only model has reduced complexity compared to the imagery-based designs. As the data is smaller, the model's complexity must decrease to avoid overfitting. This was achieved by reducing the number of weights and minimizing the number of layers. Since the feature selection algorithm permits a maximum of nine features to be utilized and imagery data is not supported, the clinical-only model can be tuned to its intended use on a greater level due to its more predictable nature. The model accesses two different machine learning components based on the number of target variables the user inputs. For a single target variable, a machine learning component has been created using the Keras Sequential API, while for more than one target variable, the component utilizes the Keras Functional API. The Functional API generally allows much more flexibility for the model architecture than the Sequential. Contrasting from the Sequential API, it allows for attributes such as shared layers and multiple outputs [Fchollet, n.d.]

Image-Clinical Model

The Image-Clinical submodel was conceived for the use of both Imagery and Clinical data. Therefore it does require a greater amount of computational expense than the Clinical-Only model. Additional functionality for this model includes the ability to load and adequately process images in the DICOM format. Given a directory that contains DICOM files, the model flattens each image into a one-dimensional NumPy array, appends each image's corresponding patient ID to the end of its array, and appends each one-dimensional array onto a two-dimensional array. This resulting array is intended to be saved to be used later for its accompanying dataset as this process can require a substantial amount of time to complete, particularly for many images. Once the completed array is obtained, it can be concatenated with its previously preprocessed clinical data by verifying the matching IDs between the two and inserting each ID's corresponding clinical data. Before utilizing the data, all IDs are removed.

The architecture of the Image-Clinical Model is similar to that of the Clinical-Only. The main difference is the greater number of layers and generally having more nodes in each layer. As the data that this submodel is intended for is much larger, its complexity should be increased.

Convolutional Neural Network

The convolutional neural network (CNN) submodel only uses imagery as input data and includes two types of layers that result in a model that functions significantly different from its counterparts: the convolutional layer and the pooling layer. Instead of the one-dimensional data utilized in the Clinical-Only and Image-Clinical models, the CNN works with two-dimensional data: non-flattened images. The convolutional layers carry out feature extraction with the use of a kernel. This kernel eventually produces a feature map by going over the data and at each

position performing matrix multiplication then summing it onto its corresponding location on the result [Cornelisse, 2018]. Several feature maps will be produced as the kernel will be replicated across different parts of the data. All of these feature maps are put together to formulate the final output of a convolutional layer. Pooling layers occur after convolutional layers and are to reduce the spatial size of the convolutional layer output. This decreases the number of parameters required to be learned following the layer; therefore, performance is generally improved [Yamashita, 2018, 615-616]. There are two main types of pooling operations: max pooling and average pooling. The model utilizes max pooling. Both variants, similarly to a convolutional layer, function using a kernel that extracts various sectors individually from the input. Max pooling outputs the maximum value in each sector and disregards all other values. Average pooling simply outputs the average of all of the values in each sector. This has a "blur" effect on data; thus, average pooling has a lessened ability to retain sharper features. Whereas the nature of max pooling is somewhat the opposite. Max pooling has a greater ability to retain sharper features and reduce the prevalence of elements such as a dark background as it will have lower values that will be discarded if there is a brighter element in the kernel. Since it is common for dark backgrounds to be prevalent in cancer imagery such as CT and MRI scans as shown in Figure 4, it is favorable to reduce noise within the data by utilizing max pooling rather than average pooling.

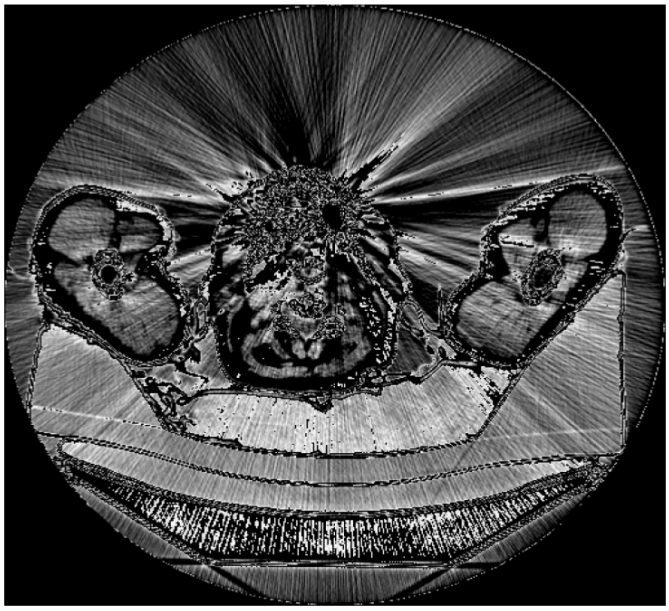

Figure 4 - CT Image from the HNSCC Dataset

After the output from the final pooling layer is obtained, it is flattened into a one-dimensional array of numbers and fed into a series of dense-layers starting with 64 nodes and a ReLU activation function as the input layer and ending with one node and a linear activation function as the output layer. Note that the CNN model does not support a multi-target regression: only one target variable at a time can be utilized.

Conclusion

The initial objective of the model was to create a computational method that could aid in compiling the most effective cancer treatment plan for a given patient. It was also an intent to create the model so that it automatically adapts to an extensive range of data. Both of these objectives have been met. Up to 91% test accuracy (n=296) has been obtained in the CNN submodel for the HNSCC-HN1 dataset with "chemotherapy_given" as the target variable with only ten epochs. This was the highest performance achieved out of all tests. Figure 5 depicts both the validation and test accuracy for the Clinical-Only model on all three datasets. Similar visualizations for the other two models can be found in the Appendix (Fig. 10 & 11). The CNN outperforms the Image-Clinical and the Clinical-Only significantly. This shows the magnitude of efficiency that the CNN possesses as it can produce higher performance than the other models with half of the epochs and with no clinical data. The most inadequate

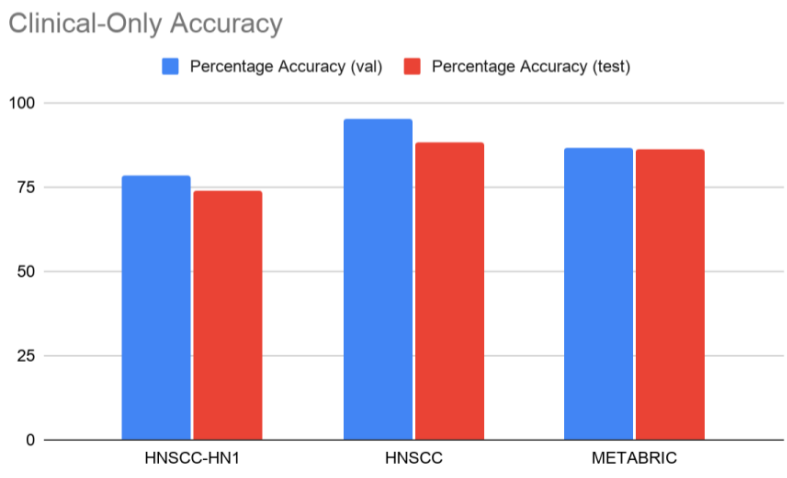

Figure 5 - Accuracy of the Clinical-Only model on all three datasets with their comparable chemotherapy variables

performance on the “chemotherapy_given” target variable was attained with the Clinical-Only model on the HNSCC-HN1 dataset with 20 epochs: 74% accuracy (n=27). With the Image-Clinical model producing 84% accuracy (n=296) with the same configuration, several conclusions can be made. First, this may show that imagery is the most effective data variant to predict this target variable. However, this is challenged with the HNSCC dataset producing its best results on the Clinical-Only model and showing on-par or worst performance with the other two models. So while this conclusion may be possible for the HNSCC-HN1 dataset, it is not valid for the HNSCC dataset. Second, the Clinical-Only model has a smaller test data size due to the lack of imagery. With only 27 examples to test with all chosen randomly and taken out of a relatively small dataset on its own (N=137), it is possible that the test data simply contained scenarios that the model was not familiar with. Note that the process used to partition the data utilizes a set random state meaning the same randomly chosen patients will be used consistently every time the model runs instead of a new set of random patients being chosen. So if this is the case, performance will not change significantly by just running the model again. From this information, it can be concluded that more clinical data from the HNSCC-HN1 dataset would be an elementary method that would likely improve the Clinical-Only model’s relatively poor performance on this dataset. Unfortunately, this cannot be done as the entire dataset has already been obtained and utilized, so a new dataset with a similar variant of data should be acquired. However, even this may not be viable in publicly available data. Other than the HNSCC-HN1 dataset on the Clinical-Only model, acceptable test performance was generally achieved on all the datasets’ binary chemotherapy variables.

In addition to the chemotherapy variable, another critical variable to test is the post-radiation therapy skeletal muscle status of a given patient, which is available only in the HNSCC dataset. This is also a binary variable that indicates either depleted or non-depleted muscle based on the skeletal muscle index calculated from post-radiation therapy imaging. The highest performance

achieved for this target variable was 71% test accuracy (n=357) on the CNN model. However, this is very close to the 70% test accuracy (n=357) achieved on the Image-Clinical model and the 69% test accuracy (n=42) achieved on the Clinical-Only model. So generally, the performance for this variable across all submodels was relatively poor. The explanations for the chemotherapy variable's performance do not apply here because there is no significant disparity between the performance of the CNN and the other two submodels, and there is a reasonable amount of testing data available.

Finally, another significant variable is within the METABRIC dataset and indicates whether a patient has received radiation therapy. 85% test accuracy (n=218) was achieved with the Clinical-Only model on this variable. This performance is acceptable. Performance data from the Clinical-Only model is presented in Table 1, and full performance details from the other two models can be seen in Tables 1 & 2 in the Appendix. In conclusion, all of the significant variables chosen produced a relatively successful accuracy. As expected, the CNN model presented higher efficiency than its counterparts and produced higher accuracy on the preeminent chemotherapy variable. Since cancer treatment is an irreversible process and every patient is unique, it is difficult to know if the accuracy achieved in the model outperforms a traditionally utilized human oncologist. Also, due to these factors, there is something to be said about the correctness of the data because, of course, it was all conceived with human judgment, and it cannot be completely verified that the patients within the datasets have been truly given the most viable treatment. Therefore, can the model outperform the humans that fabricated the datasets utilized in realistic applications? It is unclear and dependent on many factors. Nevertheless, the model is capable of at least matching

	HNSCC-HN1	HNSCC	METABRIC	HNSCC	METABRIC
Percentage Accuracy (val)	78.57	95.24	86.76	71.43	81.28
Percentage Accuracy (test)	74.07	88.1	86.24	69.05	84.86
Epochs	20	20	20	20	10
Target variable	"chemotherapy_ given"	"Received Concurrent Chemoradiotherapy?"	"chemotherapy"	"PostRT Skeletal Muscle status"	"radio_therapy"

Table 1 - Performance of the Clinical-Only model

the humans' performance in possibly a more consistent and economical way. Also, considering that the lowest test accuracy achieved was around 70%; theoretically, the highest real-world performance decrease possible would be about 30% in comparison to human oncologists. This figure may be a worthwhile tradeoff to make in return for a more financially dependable means of cancer treatment.

Further Application

More advanced variations of this kind of model, along with other machine learning cancer applications that have already been well developed, such as a diagnostic tool, could almost completely automate the cancer treatment process and significantly reduce the need for human intervention and judgment. However, the universal application of machine learning in cancer treatment presents many challenges, and more advanced versions of the model should be held to exceptionally high standards in terms of safety and inequitable bias [Kaushal et al., 2020]. Although satisfactory performance was achieved with the model, the size of some of the datasets could be questionable. Unfortunately, publicly available data can be limited. Access to non-public data could help increase the scope of situations that the model is familiar with by enlarging and diversifying the data utilized to develop it. Future studies could further compare the effectiveness of models of these types to traditionally utilized human oncologists. A clinical trial on live patients would be necessary to validate this comparison thoroughly.

Acknowledgments

The team would like to thank the New Mexico Supercomputing Challenge organization in general for arranging the resources that helped greatly in this project. Specifically, Dennis Trujillo assisted the team in the beginning stages of the project by reviewing the initial idea. Drew Einhorn assisted the team throughout the project by giving advice and answering questions, especially with the CNN model. In addition, the team would like to thank Melissa Winn for helping in preparing for presentations by arranging audiences to practice with.

References

Aerts, H., Velazquez, E., Leijenaar, R., Parmar, C., Grossmann, P., Carvalho, S., Bussink, J., Monshouwer, R., Haibekains, B., Rietveld, D., Hoebers, F., Rietbergen, M., Leemans, C., Dekker, A., Quackenbush, J., Gillies, R., & Lambin, P. (2014, June 3). Decoding Tumour Phenotype by Noninvasive Imaging Using a Quantitative Radiomics Approach. *Nature Communications*, *5*(4006). http://doi.org/10.1038/ncomms5006

Alharbi, R. (n.d.). *Breast Cancer Gene Expression Profiles (METABRIC)*. Kaggle. https://www.kaggle.com/raghadalharbi/breast-cancer-gene-expression-profiles-metabric

Cancer Treatment. (n.d.). National Cancer Institute. https://www.cancer.gov/about-cancer/treatment#:~:text=Some%20people%20with%20cancer%20will,be%20an%20option%20for%20you.

Charles, M. (n.d.). *DICOM (Digital Imaging and Communications in Medicine)*. SearchHealthIT. https://searchhealthit.techtarget.com/definition/DICOM-Digital-Imaging-and-Communications-in-Medicine

Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., Tarbox, L., & Prior, F. (2013, December). The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository. *Journal of Digital Imaging*, *26*(6), 1045-1057. https://doi.org/10.1007/s10278-013-9622-7

Cornelisse, D. (2018, April 24). *An Intuitive Guide to Convolutional Neural Networks*. Free Code Camp. https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

Elhalawani, H., Mohamed, A., & White, A. (2017). *Matched computed tomography segmentation and demographic data for oropharyngeal cancer radiomics challenges*. Sci Data 4. https://doi.org/10.1038/sdata.2017.77

Fchollet. (n.d.). *The Functional API*. Keras. https://keras.io/guides/functional_api/

Grossberg, A., Elhalawani, H., Mohamed, A., Mulder, S., Williams, B., White, A., Zafereo, J., Wong, A., Berends, J., AboHashem, S., Aymard, J., Kanwar, A., Perni, S., Rock, C., Chamchod, S., Kantor, M., Browne, T., Hutcheson, K., Gunn, G., … Fuller, C. (2020). *M.D. Anderson Cancer Center Head and Neck Quantitative Imaging Working Group*. HNSCC [Dataset]. The Cancer Imaging Archive. https://doi.org/10.7937/k9/tcia.2020.a8sh-7363

Grossberg, A., Mohamed, A., Elhalawani, H., Bennett, W., Smith, K., Nolan, T., Williams, B., Chamchod, S., Heukelom, J., Kantor, M., Browne, T., Hutcheson, K., Gunn, G., Garden, A., Morrison, W., Frank, S., Rosenthal, D., Freymann, J., & Fuller, C. (2018). *Imaging and Clinical Data Archive for Head and Neck Squamous Cell Carcinoma Patients Treated with Radiotherapy*. Scientific Data 5. https://doi.org/10.1038/sdata.2018.173

How Treatment is Planned and Scheduled. (2020, March 27). American Cancer Society. https://www.cancer.org/treatment/treatments-and-side-effects/planning-managing/planning-scheduling-treatment.html

Kaushal, A., Altman, R., & Langlotz, C. (2020, November 17). *Health Care AI Systems Are Biased*. Scientific American. https://www.scientificamerican.com/article/health-care-ai-systems-are-biased/

Lee, W. M. (2019). *Python Machine Learning*. John Wiley & Sons, Inc.

Miller, R. (2019, December 13). *Data Preprocessing: what is it and why is it important*. Ceoworld Magazine. https://ceoworld.biz/2019/12/13/data-preprocessing-what-is-it-and-why-is-important/

Population, total. (n.d.). The World Bank. https://data.worldbank.org/indicator/SP.POP.TOTL?name_desc=true

Powell, A. (2016, February 22). *The Costs of Inequality: Money = Quality Health Care = Longer Life*. The Harvard Gazette. https://news.harvard.edu/gazette/story/2016/02/money-quality-health-care-longer-life/

Ritchie, H., & Roser, M. (2019, December). *Causes of Death*. Our World in Data. https://ourworldindata.org/causes-of-death

Roser, M., & Ritchie, H. (2019, November). *Cancer*. Our World in Data. https://ourworldindata.org/cancer#:~:text=Almost%20ten%20million%20people%20die,cause%20of%20every%20sixth%20death.

Shaikh, R. (2018, October 28). *Feature Selection Techniques in Machine Learning with Python*. Towards Data Science. https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

Wee, L., & Dekker, A. (2019). *Data from Head-Neck-Radiomics-HN1 [Data set]*. The Cancer Imaging Archive. https://doi.org/10.7937/tcia.2019.8kap372n

What is Cancer? (2015, February 9). National Cancer Institute. https://www.cancer.gov/about-cancer/understanding/what-is-cancer#:~:text=Cancer%20is%20a%20disease%20caused,are%20also%20called%20genetic%20changes.

Yamashita, R. (2018, June 22). Convolutional Neural Networks: An Overview and Application in Radiology. *Springer Open*, 615-616. https://doi.org/10.1007/s13244-018-0639-9

Appendix

Figure 6

Visualization of the Pearson Correlation Coefficient for variables in the HNSCC-HN1 dataset

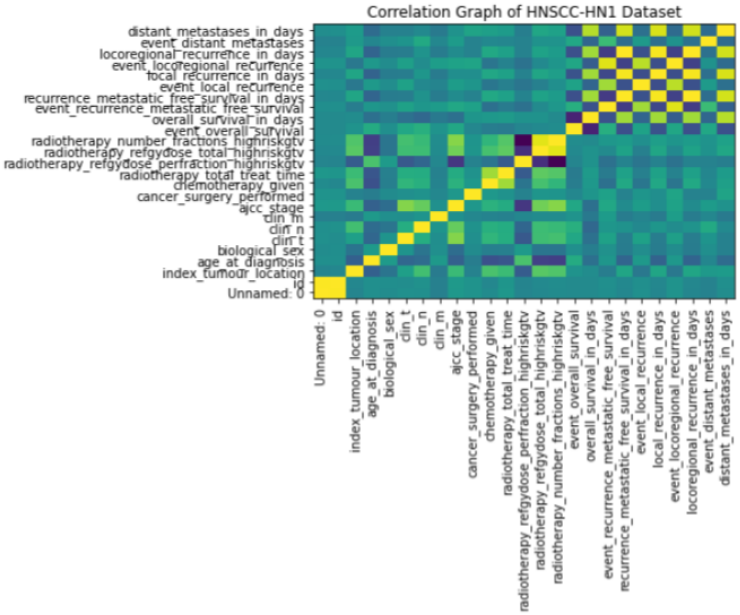

Figure 7

Visualization of the Pearson Correlation Coefficient for variables in the HNSCC dataset

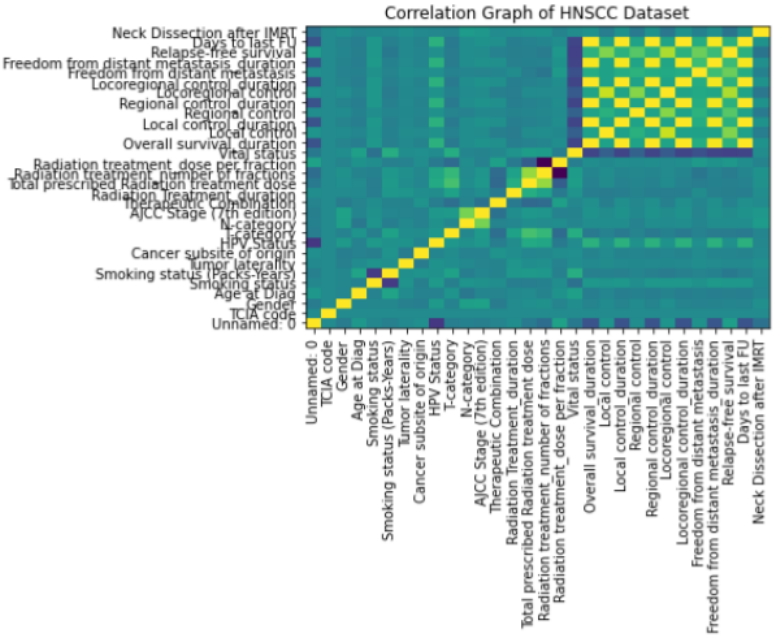

Figure 8

How the imagery of the HNSCC-HN1 dataset is utilized

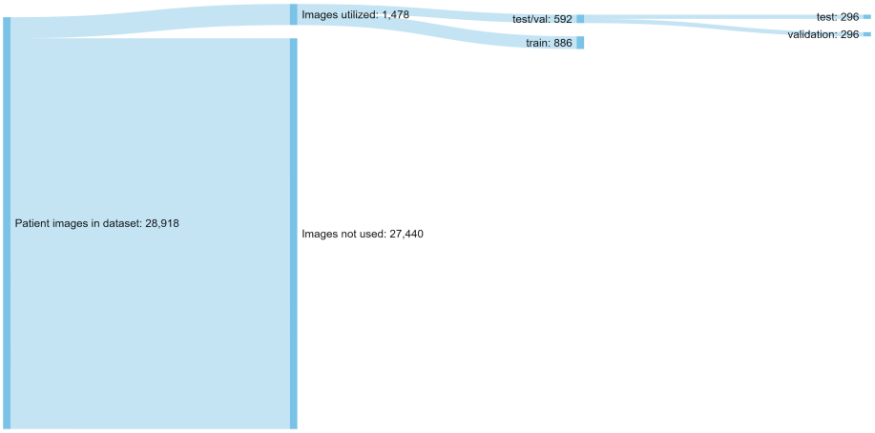

Figure 9

How the clinical data of the HNSCC-HN1 dataset is utilized

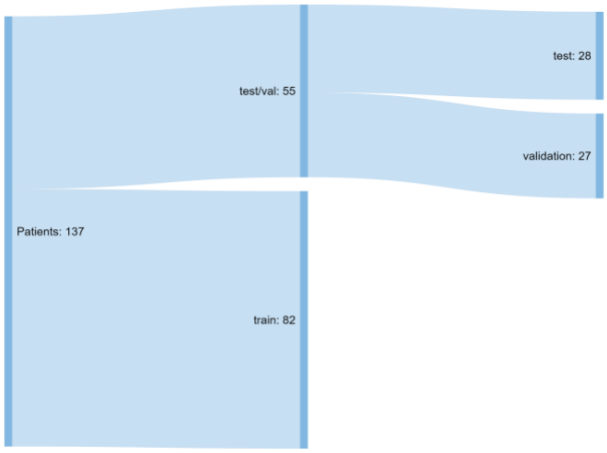

Figure 10

Accuracy of the Image-Clinical model on all datasets that incorporate imagery with their comparable chemotherapy variables

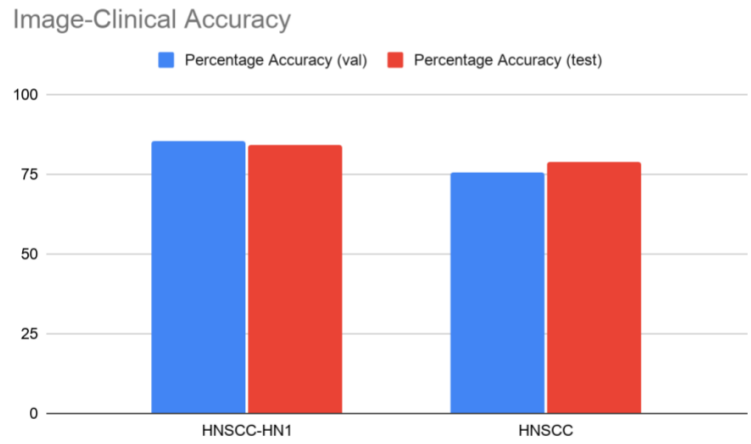

Figure 11

Accuracy of the CNN model on all datasets that incorporate imagery with their comparable chemotherapy variables

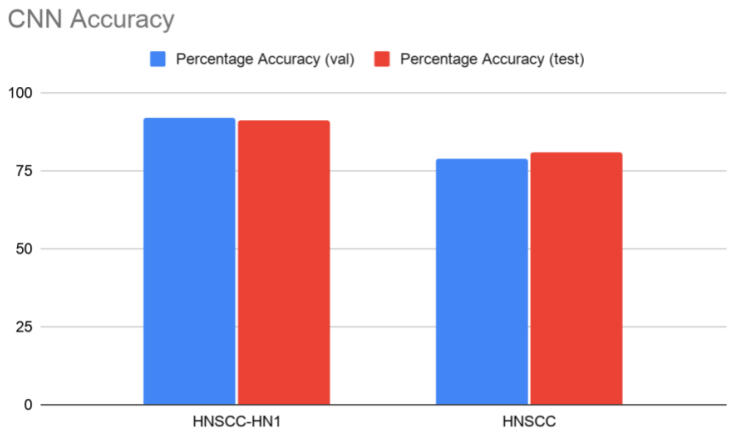

Table 2

Performance of the Image-Clinical model

	HNSCC-HN1	HNSCC	HNSCC
Percentage Accuracy (val)	85.47	75.35	75.21
Percentage Accuracy (test)	84.12	78.71	70.13
Epochs	20	20	20
Target variable	"chemotherapy_given"	"Received Concurrent Chemoradiotherapy?"	"PostRT Skeletal Muscle status"

Table 3

Performance of the CNN model

	HNSCC-HN1	HNSCC	HNSCC
Percentage Accuracy (val)	91.89	78.71	75.35
Percentage Accuracy (test)	91.22	80.95	70.59
Epochs	10	10	10
Target variable	"chemotherapy_given"	Received Concurrent Chemoradiotherapy?	PostRT Skeletal Muscle status