Improving Cervical Spine Fracture Detection Via Machine Learning

Published Blog:

https://medium.com/@eishafnu/improving-cervical-spine-fracture-detection-via-machine-learning-7d9612a60194

Previous Article:

https://medium.com/@eishafnu/detecting-cervical-spine-fractures-through-machine-learning-b8ee83636848

Previous EDA:

https://docs.google.com/document/d/1KZn7WBejTkBxu6YnCYhGDSAwp3qOXfbymmxv6WWlypI/edit#

Introduction & Motivation

Our research question is "*How can we quickly detect and locate vertebral fractures in the cervical spine from CT scans to aid in the prevention of neurologic deterioration and paralysis after trauma?"*

Despite being popular buzzwords, machine learning and artificial intelligence can and do aid in the progression of science and technology. In the healthcare sector it is increasingly being used for diagnosis, treatment, predictive analysis and personalisation of care and medicine etc.

For the purpose of this blog, we are interested in diagnosis, specifical the diagnosis of cervical spine fractures. Cervical spine fractures are essentially breaks/dislocations in the neck region and can be caused by a variety of reasons including trauma from accidents or pre-existing health conditions such as osteoporosis etc. Cervical fractures make up 56% of all spinal cord injuries (McMordie et al, 2022). The severity of such fractures differs, but the most damaging ones may lead to paralysis or even death. Given the number of total spinal cord injuries (250,000 to 500,000) and the severity of such fractures, it is essential to accurately and efficiently diagnose these cervical fractures (WHO, 2013). Our project aims to use machine learning to accurately detect cervical spine fractures, with the goal of increasing efficiency and reducing costs while matching the accuracy of a good radiologist.

Dataset

Source:
https://www.kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection/overview

Our dataset consists of approximately 2,000 CT scans of cervical spines of patients taken from 12 sites on 6 continents. The goal is to identify whether a given vertebrae has a fracture. We got the data from Kaggle as linked above.

Methodology

Data Storage Challenges

We were initially faced with the problem of space allocation as the data consists of mostly images taking up about 300GB of space. We did not have this space available on our devices and so we explored different solutions. Our solutions included finding a smaller JPEG version of the images (about 57 GB) and uploading these to Google Colab, buying an external hard-drive with 5TB of space and running the models locally, and using Kaggle notebooks to load the data directly from Kaggle. We used each option for a separate purpose. We utilized the local version of the software to closely examine the images and create videos demonstrating the progression of the skulls being overlaid on top of each other to provide a complete view of the neck. Then, we employed Google Colab for exploratory analysis of the dataset, as described in the Analysis section, and for segmentation tasks. The Kaggle platform was used to run classification algorithms to determine the likelihood of a vertebra having a fracture.

Fracture Detection Process

The fracture detection process was a multi-step process. Our first step was to use the 87 segmentation mask files and the corresponding directories to train the model for semantic segmentation. Semantic segmentation is a pixel level classification where the output corresponds to a high resolution image of the same size as the input image and assigns pixels that belong to each class based on the image (Lamba, 2019). In this case, the segmentation dataset is labeled and we are using semantic segmentation to train a model to classify what each of the vertebrae looks like. The figure in the analysis below (Figure 8) shows an image, its segmentation mask and the mask overlaid over the image. For the segmentation, we use a UNET model which is appropriate for segmentation because; it has no Dense layers meaning that images of any size can be input.

The second step is to train a classification model to identify the fractures. Given the complexity of the dataset, the difficulty in identifying fractures and the great class imbalance, we went straight to deep learning for the classification tasks. Guided by our research, we went with a selection of 4 models:

- a basic CNN
- three pretrained transfer learning models:
    1. InceptionV3
    2. ResNet50
    3. EfficientNetB5

Each of the pretrained models was chosen based on research about the most commonly used models in medical image classification and especially when dealing with fractures (Emon et al., 2022). We also considered the efficiency metrics detailed in the Kaggle website in terms of size of the model, time it takes to run among others.

CNN

The simple CNN utilizes a sequential model with multiple hidden convolution blocks and artificial Dense blocks as part of the output layers.

Advantage:

CNN is that it has fewer layers and does not rely on transfer learning, making it faster compared to other models.

Disadvantage:

It needs to learn the weights from scratch, which means it has a low starting point and a higher bias towards the specific data, resulting in less variance and a higher risk of overfitting the data.

Pertained Transfer Learning Models

Transfer learning is used when employing these models, meaning that they have been trained on millions of diverse images and have had their weights optimized.

Advantage:

Using transfer learning allows for more efficient training and weight updates, as the model has already been exposed to a diverse set of images, reducing the risk of overfitting. Additionally, the familiarity with image datasets enables the model to more easily recognize images.

Disadvantage:

These models are often very deep, with layer counts ranging from 16 to over 500, which results in longer training times compared to the basic CNN, especially without GPU acceleration.

Therefore, when choosing the models to try, applicability and efficiency are very important considerations. Out of the three models, ResNet50 has the shortest depth, making it relatively fast but not as reliable as EfficientNetB5. For all of these models, we only trained them for 10 epochs and used the validation loss for early stopping if the loss was not improving. Ideally, it would be beneficial to run the models for at least 20 epochs to give them more time to learn, but due to the time required (about 12 minutes per epoch) and the lack of GPU acceleration, we determined that 10 epochs was sufficient for making predictions and comparing the models.

Next, for the threshold metrics, we carefully considered metrics that would deal with the high class imbalance identified in the analysis section. F1 score and ROC score helps tell us whether the models are able to overcome the issue with class imbalance. The F1 score is the harmonic convergence of the recall and precision. Precision indicates the fraction of correctly predicted positive outcomes, while recall reflects the ability to accurately predict the positive class. AUC score calculates the false positive and false negative rate and gives a value between 0 and 1 where a value near 1 means the classifier does a good job distinguishing between the positive and negative classes while near 0.5 means it is no better than random and anything below that is not useful (Brownlee, 2020).

Analysis

Now we will provide an exploratory analysis of the metadata, images and segmentation files.

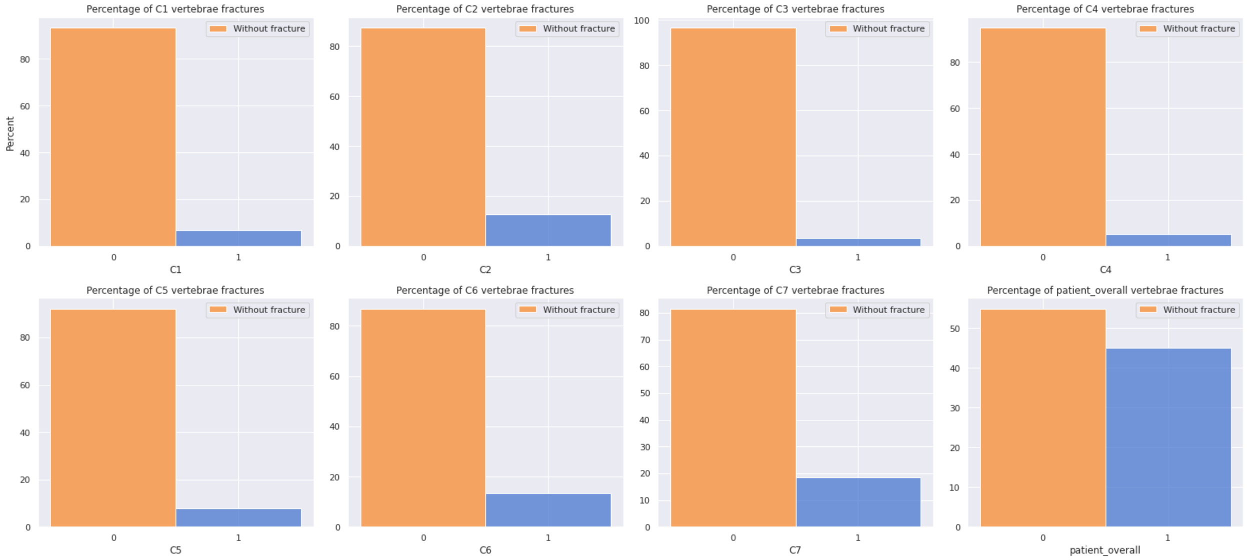

Figure 1: Percentage of fractures in each of the vertebrae and in the overall patient.

From Figure 1, we can see that C1, C3, C4, C5 vertebrae have the lowest proportion of fractures. On the other hand, we can see that C2, C6, and C7 have comparatively higher proportions of fractures. C7 has the highest proportion of fractures compared to other vertebrae because its the vertebrae that connects the neck with the upper-back and is highly associated with back injuries. Back injuries are much more common than neck injuries which explains this difference. C3 and C4 have particularly low fractures because of the curvature of the neck which means that they are protected by the other vertebrae in the event of a neck injury. Figure 2 below shows the total number of fractures per vertebrae ordered lowest to highest. It’s a more compact side by side comparison that has the same insights as Figure 1.

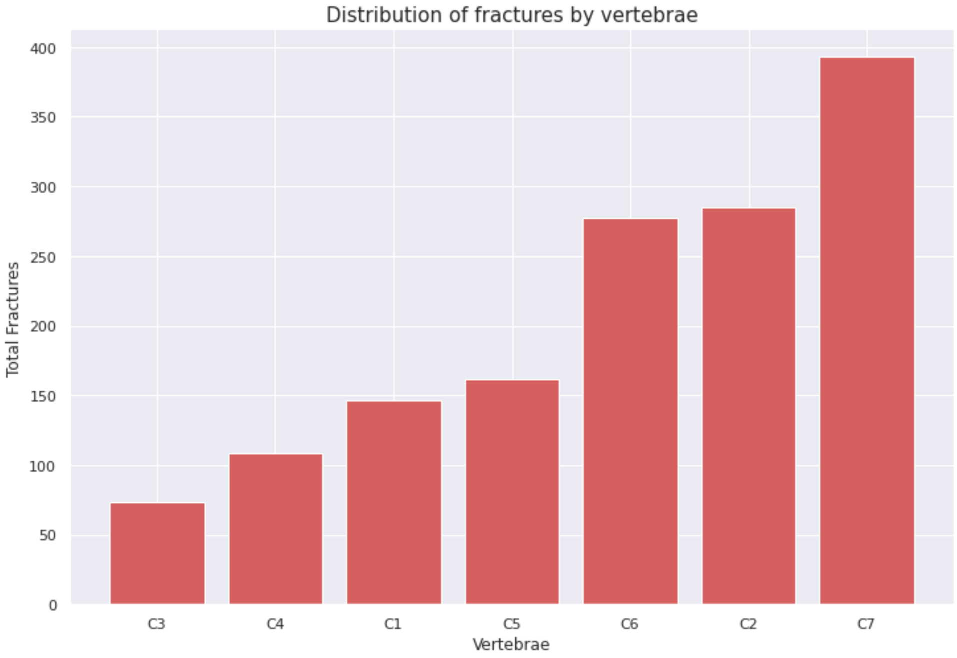

Figure 2: Total number of fractures per vertebrae across all studies

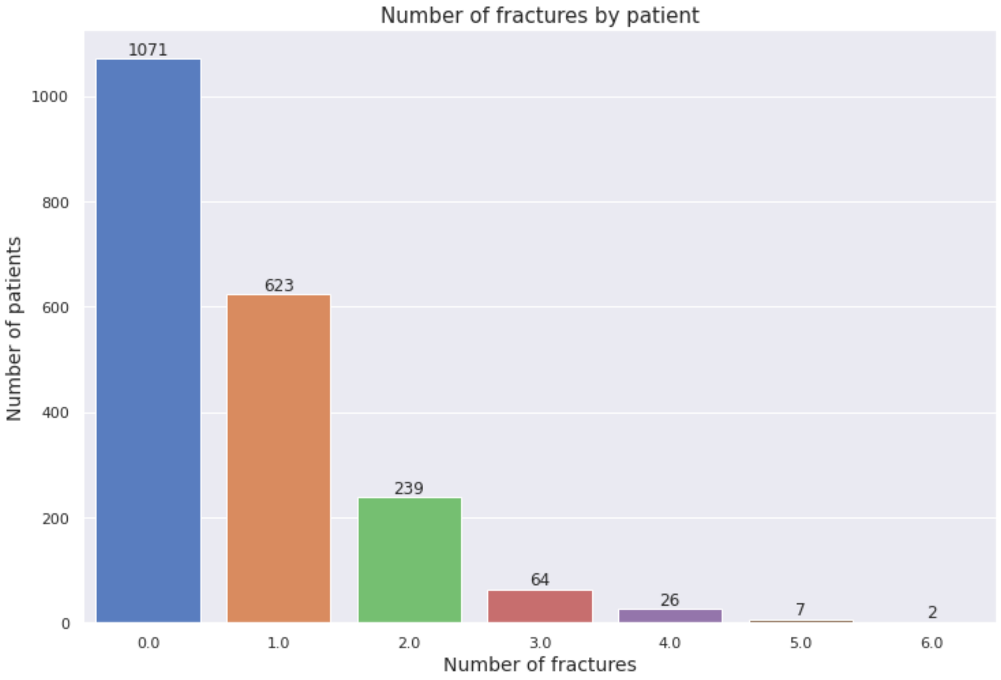

Figure 3: Number of fractures by number of patients (distribution of fractures by patients)

By looking at the distribution of the fractures by patients, we can see that most patients do not have fractures. That is out of the 2019 studies, about half of them do not have a single fracture while the remaining half has at least one fracture with majority of patients having only one fracture. It's very rate to have 5 or more fractures as chances of survival with that many fractures are very low. A good proportion of patients (about 10%) have two fractures which makes sense as seen in the correlation plot in Figure 4 below.

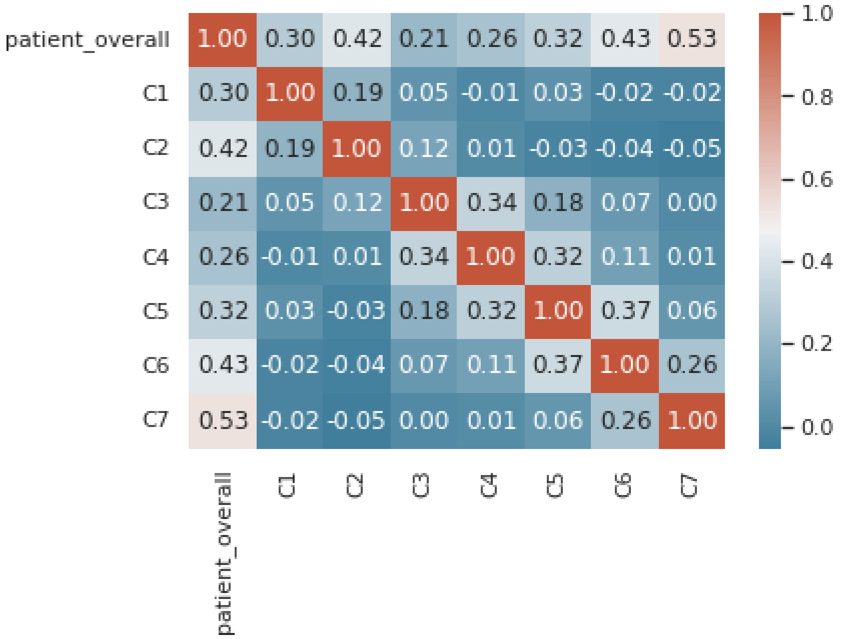

Figure 4: Correlation plot showing the relationship between the vertebrae

The analysis shows a strong correlation between C7 and the overall patient status. This is expected, as C7 is the vertebra with the highest number of fractures among patients. Therefore, if a patient has a fracture in C7, they are likely to have fractures overall. However, what is interesting is the moderate level of correlation between C3 and C4, C4 and C5, and C5 and C6. This makes sense, considering their proximity to each other. If there is an injury in C3, there is a likelihood of an injury in C4 as well.

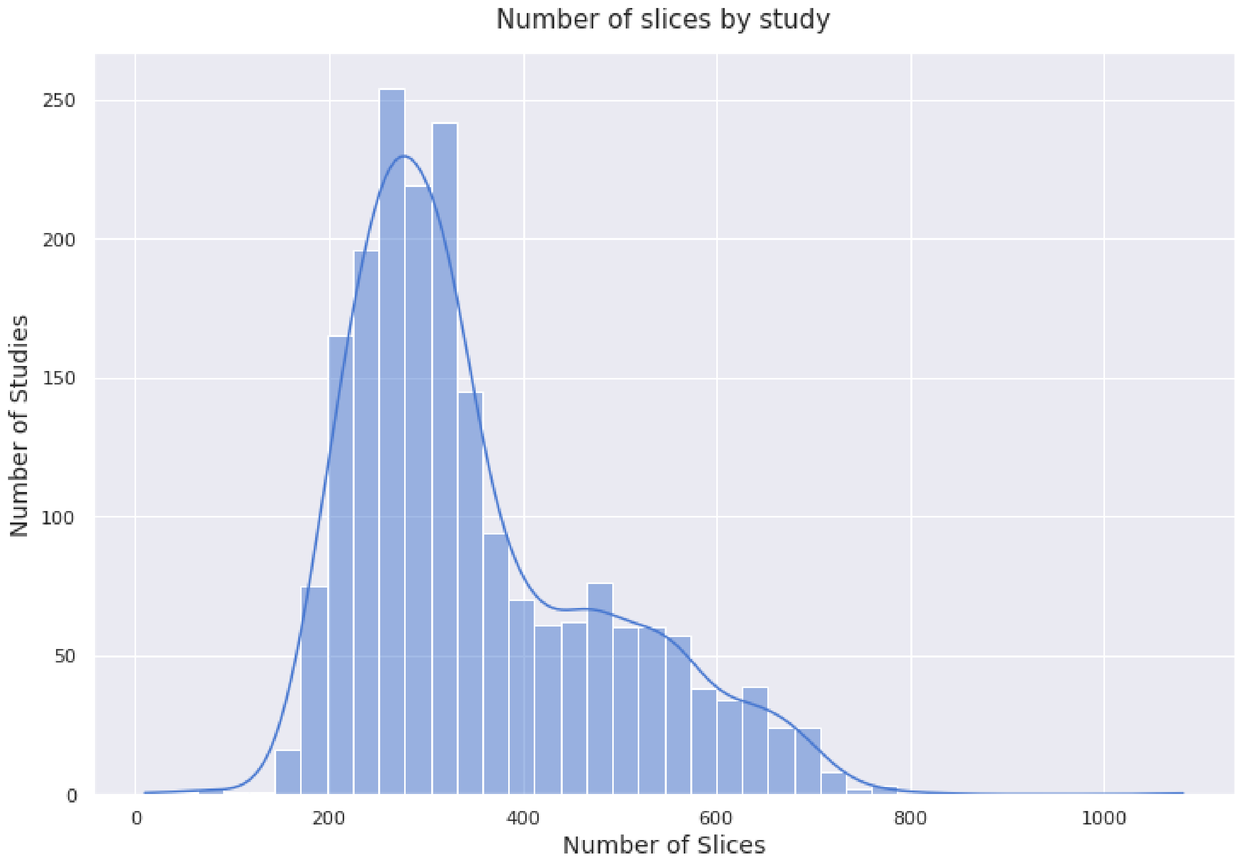

Figure 5: Distribution of slices by studies.

The distribution of slices across studies has a slight bimodal distribution and a right-skew. The median number of slices per study is 314 while the average number of slices per study is 352. This tells us that on average, there are about 350 slices in a given study and therefore we can expect sufficient number of images to train the model despite the fact that there are only 2019 total studies provided.

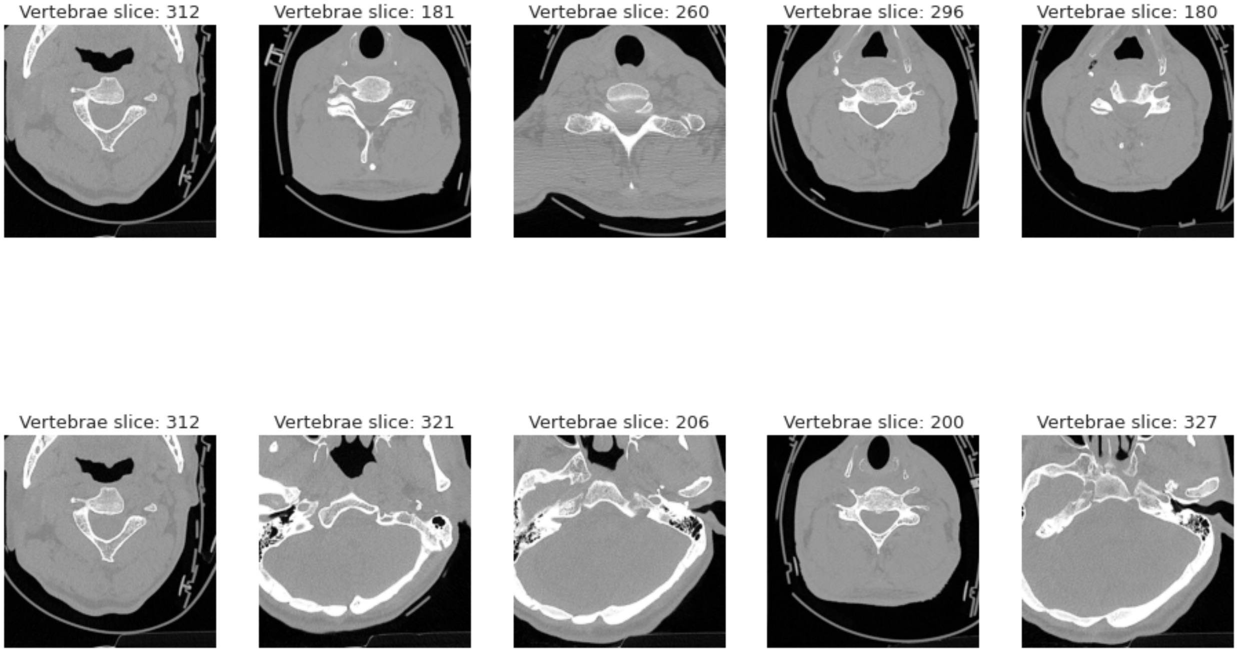

Figure 6: Randomly selected top-level slices of the cervical spine.

By selecting random images from one of the studies, it is possible to see what the scans look like and begin to identify the cervical vertebrae, as shown in Figure 6. The slices represent approximately 1mm of images taken during the CT scan, and overlaying them produces a complete picture of the entire scan from the base of the skull to the upper back vertebrae, as demonstrated in the video. Some of these images may contain fractures, but it can be difficult to determine this without the expertise of a radiologist. Additionally, it can be challenging to determine which images correspond to which vertebrae without prior knowledge. The segmentation masks shown in Figure 7 can assist with this process.

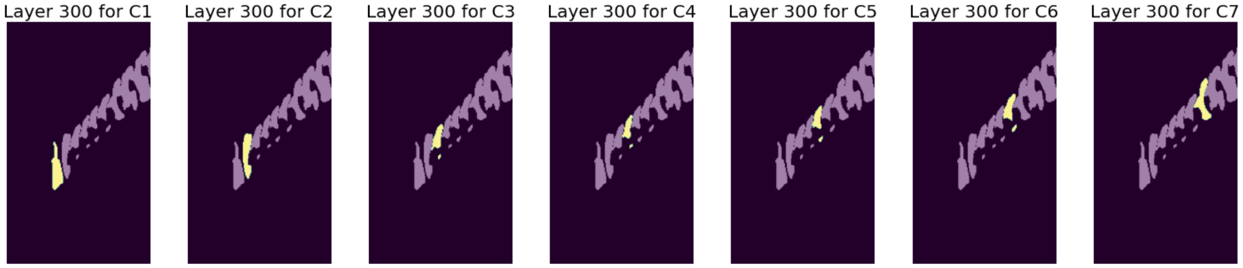

Figure 7: Side view of the neck showing the position of each of the vertebrae based on segmentation masks.

Figure 7 shows the segmentation masks from a slice of a layer of an image. These segmentations represent where each pixel of each vertebrae is where yellow is the given vertebra and purple is all the other vertebrae. A clearer picture of the usefulness of segmentation is shown below.

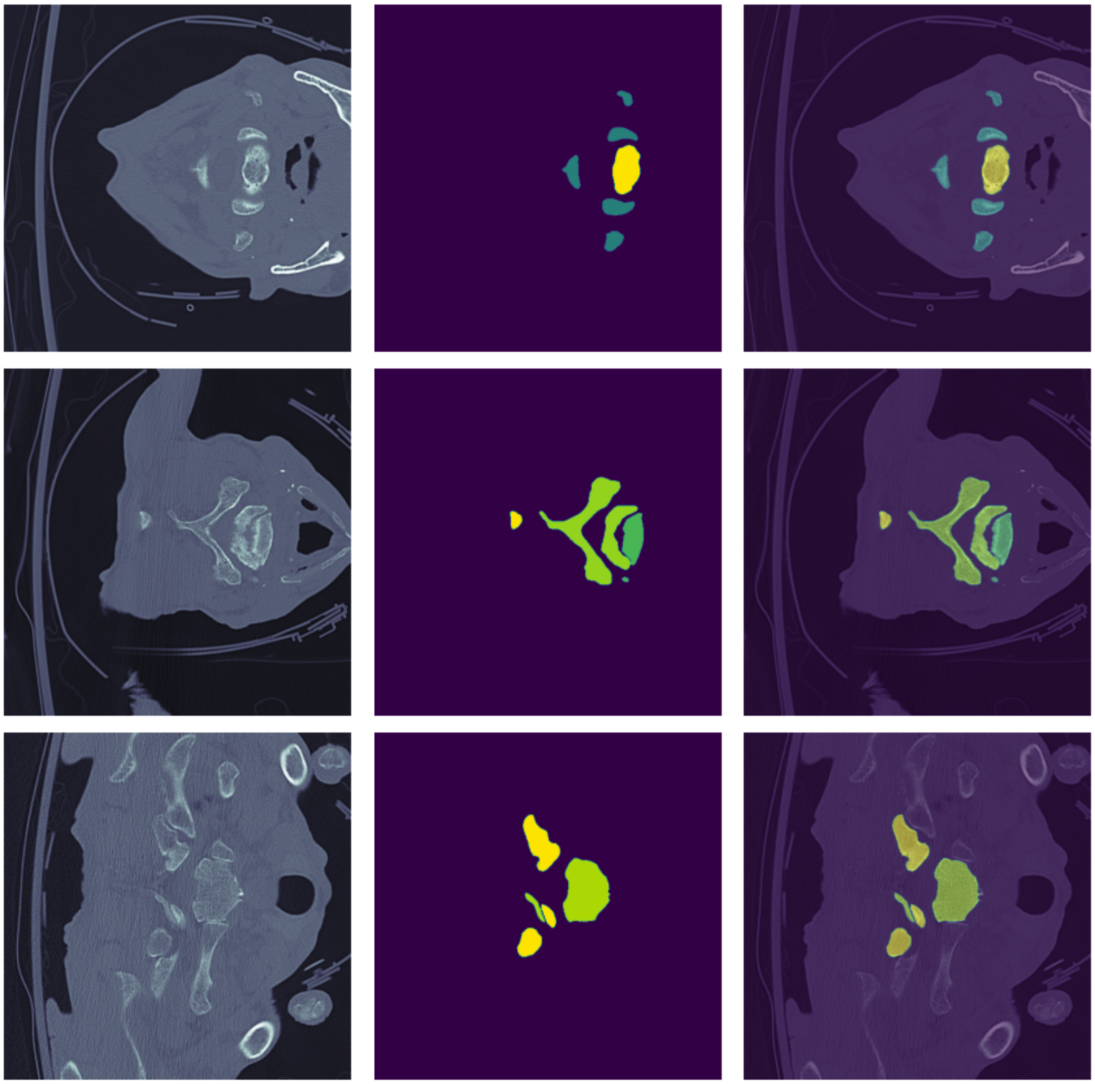

Figure 8: Random vertebrae images, the corresponding mask and the image overlaid by the mask

Figure 8 demonstrates the importance of image segmentation. The masks in the middle column have labels, with each color representing a different vertebra. The left column corresponds to the unlabeled image slices, and the right column shows the image slices overlaid with the mask to identify the vertebrae present in the image based on the labeled mask. A video of the masks overlaid on the images can be found here. By using this dataset, it is possible to train a model to recognize the appearance of different vertebrae and make predictions on unseen vertebrae.

Results

The UNET model did not produce any results. The first epoch took over an hour to complete, and when it did, the result was zero accuracy and zero loss. This suggests that the model would require a large number of epochs to generate somewhat reliable results. It is also possible that the data and model were configured incorrectly, leading to the model processing the input incorrectly and resulting in zero accuracy. However, the classification model can still learn to identify the types of fractures without the segmentation module, as it will be provided with similar-looking vertebrae, some of which have fractures and some of which do not. Eventually, the classification model will learn to recognize C1 vertebrae with fractures and predict their probability. While the segmentation model is important in the long term to improve the performance of the classification model and make it easier for radiologists to identify the C1-C7 vertebrae, it is not necessary for the classification model to function.

To our surprise, the basic CNN had the best performance besides having the lowest number of layers. It had the lowest loss and highest F1 and AUC scores. Additionally, it was also the fastest model to run and therefore the best overall. One possible explanation for the unexpected result is that the researchers had a better understanding of how to configure a basic CNN model and were able to optimize the configuration to suit the needs of the images. On the other hand, the pretrained models have predetermined weights and are more difficult to customize, as a thorough understanding of the underlying processes is necessary.

Another potential reason could be that the pretrained models are very deep, with numerous layers, and require at least 50 epochs to improve performance. However, as previously mentioned, the runtime for each epoch is too lengthy without GPU acceleration, so we wanted to see how each model would perform with minimal tuning.

The accuracy and loss models shown in Figures 9 to 12 paint a good picture of the performance of the models.

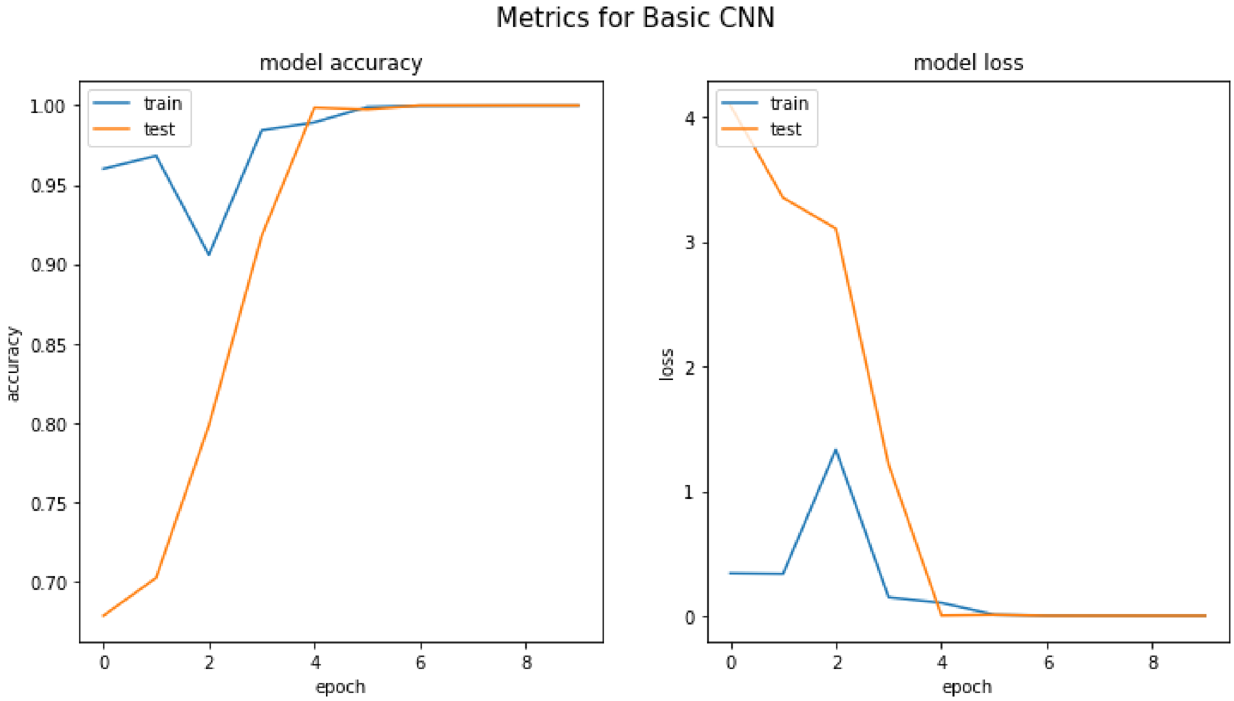

Figure 9: Accuracy and loss for Basic CNN with 10 epochs

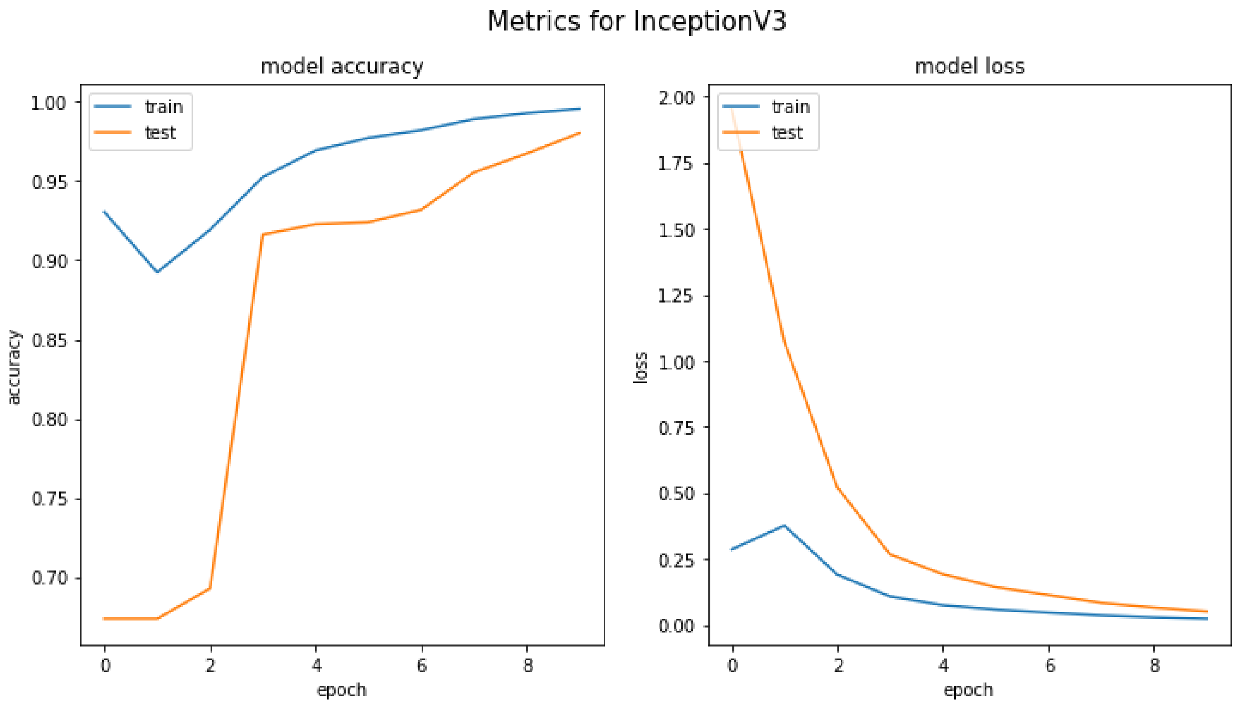

Figure 10: Accuracy and loss for InceptionV3 with 10 epochs

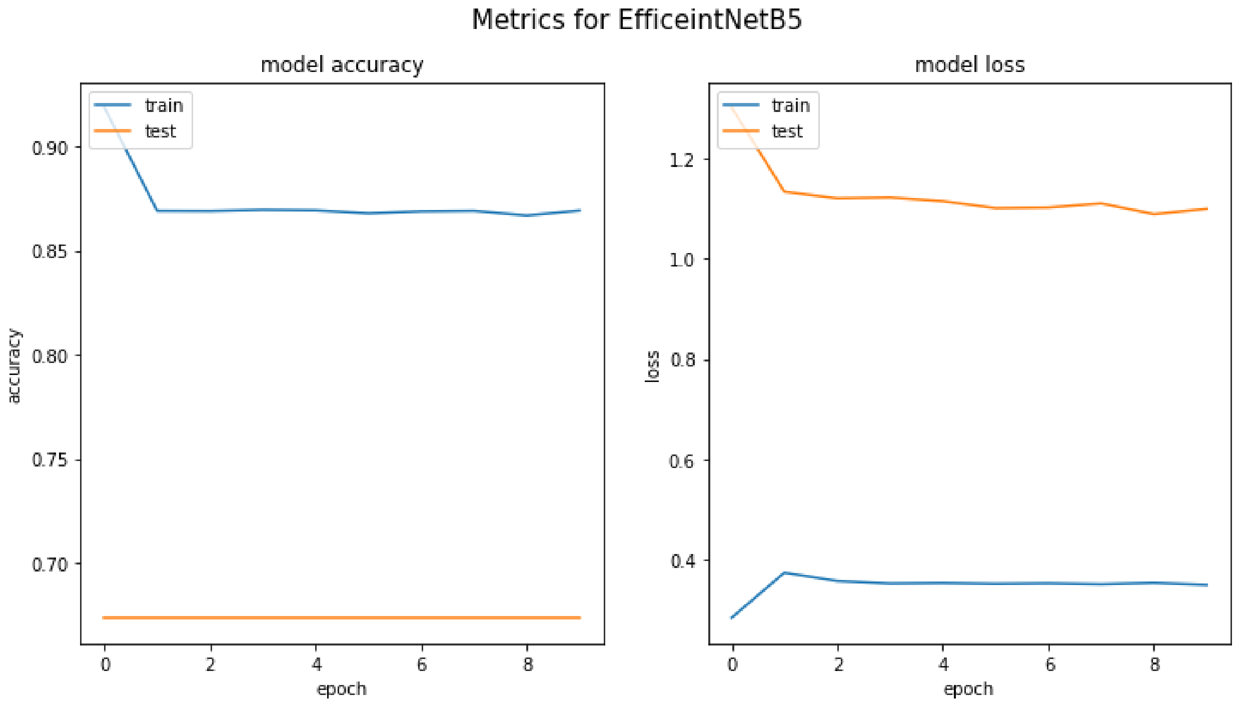

Figure 11: Accuracy and loss for ResNet50 with 10 epochs

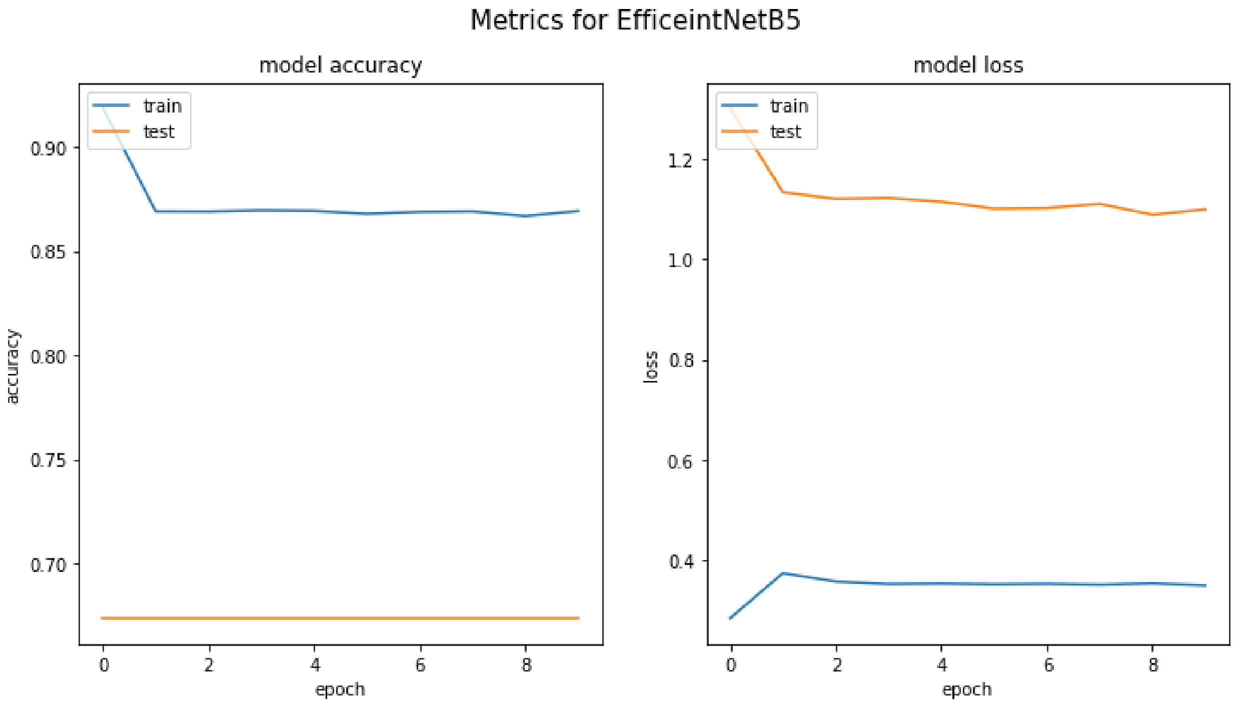

Figure 12: Accuracy and loss for EfficientNetB5 with 10 epochs

The lower the loss, the better the model will perform in classification. The high AUC of the basic CNN tells us that the model has a low false positive and false negative rate and the high F1 score tells us it does a very good job differentiating between fractured vertebrae and those without fractures but the . This makes it a very reliable model especially since we want to avoid false negatives ie we do not want to misclassify the patient as not having a fracture when they actually have a fracture. We also want to reduce false positives because we do not want patients to undergo expensive treatment for fractures they did not have.

rently to achieve better results.

After determining the basic cnn had the best performance, we evaluated the model on 5 directories of unseen test data to predict the probability that a given vertebrae has a fracture. The results of these predictions are summarized in Table 4. Ideally, we should have improved on this model using k-fold cross-validation but training time and resources were not in our favor.

The results show that the basic CNN model does a very good job in predicting the probability that a given vertebrae has a fracture. This can be seen by comparing Table 3 which shows the true values and Table 4 which shows the predicted values.

Conclusion

Although we were not able to successfully get results from the UNET segmentation model, we were able to set it up for training and get results from one epoch. A future endeavor is getting the model to be more efficient and to configure it correctly so that it can run at least 10 epochs. With this model, we can combine it with the classification model to help radiologists determine which vertebrae have fractures.

From trying out different models, the best performing model turned out to be the basic CNN model. As mentioned, there is room for improvement in the model using cross-validation but the metrics are already very impressive. The model also handled the class imbalance very well as measured by the AUC and F1 sores.

The combination of the classification and the segmentation models would save radiologists a lot of time because all they would need to do is given the model a scan and it would return a prediction of they type of vertebrae and which if any have fractures. The radiologists would not need to go through an average of 350 images per patient and would focus their efforts on diagnosis and treatment. To make their work even easier, an extension of this would be to create a website where radiologists simply upload DICOM files and they get back the predicted results. The cleaning and pre-processing steps for the DICOM would be very similar to what we have did. As radiologists add data, the data could be used to train the models even more to increase their accuracy while avoiding misdiagnosis.

Github Repo

Main Github: https://github.com/Hadavand-s-Minions/rsna-cervical-spine

Analysis Notebooks:
https://github.com/Hadavand-s-Minions/rsna-cervical-spine/tree/main/notebooks

References

Exploratory Analysis Assignment: https://docs.google.com/document/d/1KZn7WBejTkBxu6YnCYhGDSAwp3qOXfbymmxv6WWlypI/edit#heading=h.up2wakf2yq2g

Accuracy of the Canadian C-spine rule and NEXUS to screen for clinically important cervical spine injury in patients following blunt trauma: A systematic review—PMC. (n.d.). Retrieved December 16, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3494329/

Blackmore, C. C., Ramsey, S. D., Mann, F. A., & Deyo, R. A. (1999). Cervical Spine Screening with CT in Trauma Patients: A Cost-effectiveness Analysis. *Radiology*, *212*(1), 117–125. https://doi.org/10.1148/radiology.212.1.r99jl08117

Brownlee, J. (2020, January 7). Tour of Evaluation Metrics for Imbalanced Classification. *MachineLearningMastery.Com*. https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

Chauhan, S., Vig, L., De Filippo De Grazia, M., Corbetta, M., Ahmad, S., & Zorzi, M. (2019). A Comparison of Shallow and Deep Learning Methods for Predicting Cognitive Performance of Stroke Patients From MRI Lesion Images. *Frontiers in Neuroinformatics*, *13*. https://www.frontiersin.org/articles/10.3389/fninf.2019.00053

Emon, M. M., Ornob, T. R., & Rahman, M. (2022a). Classifications of Skull Fractures using CT Scan Images via CNN with Lazy Learning Approach. *Journal of Computer Science*, *18*(3), 116–129. https://doi.org/10.3844/jcssp.2022.116.129

Emon, M. M., Ornob, T. R., & Rahman, M. (2022b). Classifications of Skull Fractures using CT Scan Images via CNN with Lazy Learning Approach. *Journal of Computer Science*, *18*(3), 116–129. https://doi.org/10.3844/jcssp.2022.116.129

Emon, M. M., Ornob, T. R., & Rahman, M. (2022c). Classifications of Skull Fractures using CT Scan Images via CNN with Lazy Learning Approach. *Journal of Computer Science*, *18*(3), 116–129. https://doi.org/10.3844/jcssp.2022.116.129

Guermazi, A., Tannoury, C., Kompel, A. J., Murakami, A. M., Ducarouge, A., Gillibert, A., Li, X., Tournier, A., Lahoud, Y., Jarraya, M., Lacave, E., Rahimi, H., Pourchot, A., Parisien, R. L., Merritt, A. C., Comeau, D., Regnard, N.-E., & Hayashi, D. (2022). Improving Radiographic Fracture Recognition Performance and Efficiency Using Artificial Intelligence. *Radiology*, *302*(3), 627–636. https://doi.org/10.1148/radiol.210937

Image segmentation using UNet. (n.d.). Retrieved December 16, 2022, from https://kaggle.com/code/mineshjethva/image-segmentation-using-unet

Kumar, D. V. (2020, June 19). *Implementing EfficientNet: A Powerful Convolutional Neural Network*. Analytics India Magazine. https://analyticsindiamag.com/implementing-efficientnet-a-powerful-convolutional-neural-network/

Kumar, Y., & Hayashi, D. (2016). Role of magnetic resonance imaging in acute spinal trauma: A pictorial review. *BMC Musculoskeletal Disorders*, *17*, 310. https://doi.org/10.1186/s12891-016-1169-6

Kurama, V. (2020, June 5). *A Guide to ResNet, Inception v3, and SqueezeNet*. Paperspace Blog. https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/

Lamba, H. (2019, February 17). *Understanding Semantic Segmentation with UNET*. Medium. https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47

Maynard-Reid, M. (2022, February 21). U-Net Image Segmentation in Keras. *PyImageSearch*. https://pyimagesearch.com/2022/02/21/u-net-image-segmentation-in-keras/

Munera, F., Rivas, L. A., Nunez, D. B., & Quencer, R. M. (2012). Imaging Evaluation of Adult Spinal Injuries: Emphasis on Multidetector CT in Cervical Spine Trauma. *Radiology*, *263*(3), 645–660. https://doi.org/10.1148/radiol.12110526

Pretrained Models for Transfer Learning in Keras for Computer Vision. (n.d.). DEV Community 👩‍💻👨‍💻. Retrieved December 16, 2022, from https://dev.to/amananandrai/pretrained-models-for-transfer-learning-in-keras-for-computer-vision-5eei

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (arXiv:1505.04597). arXiv. http://arxiv.org/abs/1505.04597

Rosebrock, A. (2017, March 20). ImageNet: VGGNet, ResNet, Inception, and Xception with Keras. *PyImageSearch*. https://pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/

RSNA 2022 Cervical Spine Fracture Detection. (n.d.). Retrieved December 16, 2022, from https://kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection

Small, J. E., Osler, P., Paul, A. B., & Kunst, M. (2021). CT Cervical Spine Fracture Detection Using a Convolutional Neural Network. *AJNR: American Journal of Neuroradiology*, *42*(7), 1341–1347. https://doi.org/10.3174/ajnr.A7094

Spinal cord injury. (n.d.). Retrieved December 16, 2022, from https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury

Tan, M., & Le, Q. V. (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* (arXiv:1905.11946). arXiv. http://arxiv.org/abs/1905.11946

Team, K. (n.d.). *Keras documentation: Keras Applications*. Retrieved December 16, 2022, from https://keras.io/api/applications/

Themes, U. F. O. (2019, August 20). Spinal Trauma. *Radiology Key*. https://radiologykey.com/spinal-trauma-4/

Tutorial Keras: Transfer Learning with ResNet50. (n.d.). Retrieved December 16, 2022, from https://kaggle.com/code/suniliitb96/tutorial-keras-transfer-learning-with-resnet50