Cervix Spine Fracture Classifier using Enhanced Canny Edge Detection and Support Vector Machine

John Markton M. Olarte[1,2], Limuelle D. Alamil[1,3], Emmanuel Tarek Shayne Guides[1,4], and Ara Abigail E. Ambita[1,5]

[1] University of the Philippines Visayas
[2] jmolarte@up.edu.ph
[3] lcalamil@up.edu.ph
[4] eguides@up.edu.ph
[5] aeambita@up.edu.ph

Abstract. This mini-project aimed to develop a classification model using Support Vector Machine (SVM) to accurately predict cervical spine fractures in x-ray scans. The group collected data from Kaggle, applied image pre-processing *(using Enhanced Canny Edge Detection as a feature selection technique)* and data pre-processing techniques, and trained three different SVM models. The models were evaluated using the confusion matrix, classification report scores, and AUC-ROC curve, with SVM Model (b) performing the best in terms of accuracy, precision, recall, and F1 scores. However, SVM Model (a) had the fastest run-time, followed by SVM Model (c) and SVM Model (b). Overall, these results suggest that machine learning and computer vision techniques can improve the accuracy and efficiency of cervical spine fracture diagnosis.

Keywords: Cervix Spine Fracture · SVM · Canny Edge Detection

6 7

[6] Mini Project's GitHub Repository: bit.ly/CervixSpineClassifier
[7] Cervix Spine X-ray Images Dataset Available From: bit.ly/KaggleCSF_dataset

1 Introduction

1.1 Background and Rationale

The cervical spine commonly referred to as the neck region consists of seven bones, known as vertebrae, that support and connect the head to the shoulders. At the center of these vertebrae is the spinal cord which is the nervous system's connection between the brain and body. Any break or injury in one of these bones due to high-energy trauma or shock results in a condition called cervical spine fracture [1]. To evaluate a cervical spine fracture, plain X-ray observations are usually adequate, however, in cases of minimally fractured bones (i.e., instabilities that are more difficult to observe), computed tomography (CT) scans are required. A CT scan is considered the best test to verify the presence of a fracture in the cervical spine [2].

Nonetheless, cervical spine fractures are still likely to go undetected by doctors, especially during its initial phase [3][4]. This poses a major problem considering that cervical spine fractures may damage the spinal cord and lead to serious complications.

Specifically, the following are different consequences that arise from cervical spinal fractures [5]:

1. Discomfort around the neck region as well as reduced neck motion.
2. Pain with varying intensity that radiates from the neck into the shoulder, arms, and fingers.
3. Difficulty in breathing.
4. Muscle weakness brought about by partial or total paralysis.
5. Numbness of sensory nerves.

In essence, there is a need to increase the likelihood of a successful clinical diagnosis and detection of cervical spine fracture in the field of radiology which is the primary motivation of this mini project. More specifically, the developers of this mini project aim to create a classification model that can help assist radiologists in early diagnosis of cervical spine fractures.

About RSNA 2022 AI Challenge

This mini-project was inspired by the 2022 Radiological Society of North America (RSNA) Cervical Spine Fracture AI Challenge. According to their data, out of the estimated 1.5 million vertebral compression fracture cases in the U.S., only 25-33% of radiographically identified vertebral fractures are clinically diagnosed [6]. Consequently, the challenge aims to quickly detect and determine the location of vertebral fractures through the development of machine learning models that can match the performance of actual radiologists.

1.2 Objectives

General Objective

To develop a binary classification model using Enhance Canny Edge Detection (ECED) and Support Vector Machine (SVM) to classify whether a cervical X-ray scan is fractured or not.

Specific Objectives

1. To create different models using different SVM kernels.
2. To use different metrics in order to gauge and compare the performance of the different models.
3. To compare the run-times of different models for classifying cervix spine radiological images.

1.3 Review of Related Works and Literature

Radiological Image Feature Selection

Pre-processing the data gathered is one of the most crucial steps in every machine learning algorithm and its applications. In processing images, especially those that are scans from X-rays, one of the most rudimentary techniques is edge detection, in which discontinuities in the brightness of an image are highlighted [7].

As stated in the paper titled "Enhanced Canny edge detection for COVID-19 and pneumonia X-ray images" by Hwa, Bade, and Hijazi (2020), Canny Edge detection prevails as the preferred edge detection algorithm since it is more resilient against noise and produces sharper and clearer edges. With these promising baselines for an image processing algorithm, the authors have proceeded to implement some improvements to Canny Edge detection, labeling the newer algorithm as CEED-CANNY. In the testing done by the authors, CEED-CANNY has produced smaller MSEs (mean squared errors) when compared to the traditional Canny edge detection algorithm when applied to detecting COVID-19 or pneumonia on some sample X-ray scans [8].

As this paper is utilizing the Canny Edge Detection algorithm for cleaning and pre-processing the scans, which if not processed, can be disorienting as the noise in some, if not most, of them is excessive, the paper written by Hwa et al. (2020) has given some insights on how exactly the algorithm works and has even promoted a slightly improved version of it.

Application of Machine Learning Models for Radiological Image Classification

In the paper by Li et al. (2012), the K-Nearest Neighbor algorithm, which is among the simplest of all machine algorithms [9], was used in the classification of lymph node metastasis in gastric cancer. It was stated that machine learning methods, as mentioned beforehand, were applied to the Gemstone Spectral Imaging of lymph node metastasis in order to provide more information regarding the issue, of which, has actually gained some popularity in the field of clinical medical analysis. After the retrieval of images or scans to be used for the preprocessing and classification from the Gemstone Spectral Imaging (GSI) system, feature selection was then done in order to reduce feature space and data dimensions. The K-nearest neighbor classifier was then implemented in order to classify lymph node metastasis against those that are non-lymph node metastasis. The methods used showed a promising accuracy of 96.33% over 38 samples [10].

Studying this paper has helped gather insight onto the researchers as it has proven that other machine learning classification algorithms, which in this case is the K-nearest neighbors classifier, is just as qualified and utilized in the field of biomedical sciences.

1.4 Project Expectations

To be able to develop at least one model based on SVM that can correctly assess and classify Cervical Spine Fracture on cervical X-ray scans, in a reasonable amount of time, with at least 85% accuracy in its prediction. Since, as mentioned previously by the Radiological Society of North America, only 25–33% of all vertebral fractures are clinically diagnosed, having the models overtake that statistic by 50% is within expectations. Additionally, based on the paper by Nitze et. al. (2012) about the speed of some machine learning algorithms, including SVM, when it comes to classifying crop types, SVM with a Radial Basis Function (RBF) kernel took an average of 0.035 seconds in the classification of one image, whereas the SVM with a Polynomial Kernel took 0.018 seconds [11]. With this in mind, having an expectation of around two to three images being classified within a single second would be reasonable.

2 Methodology

In this mini-project the team created SVM models for classifying between fractured and non-fractured cervical spine by following the figure in the succeeding page.

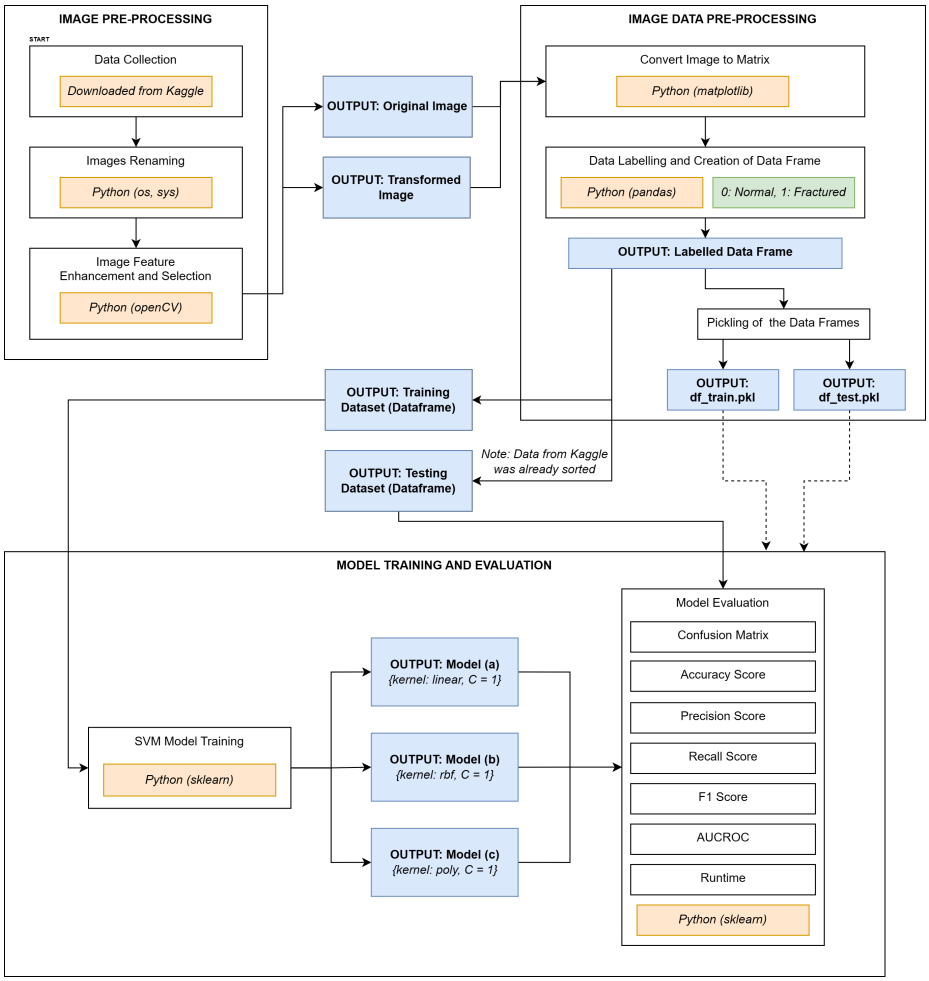

Fig. 1. Enhanced Canny Edge - Support Vector Machine Flowchart

Figure 1 shows the processes followed in the conduct of this mini-project. Which can be seen divided into three main processes, namely: (1) Image Pre-Processing, (2) Image Data Pre-Processing, and (3) Model Training and Evaluation.

2.1 Image Pre-Processing

Image pre-processing was done primarily to (1) collect the cervix spine radiological scans from Kaggle, (2) rename the image files, and (3) enhance and select the feature needed for the SVM model training and evaluation.

Data Collection

The cervical spine image dataset was downloaded from Kaggle. This dataset is composed of images gathered from the RSNA cervical spine fracture detection challenge. The dataset has an overall uncompressed size of 354.63 MB. This, however, is already expected as it contains 4,200 images of scans, split between the training set (3,800 images) and the testing set (400 images) upon downloading. The training and testing set were also subdivided into two directories that contain the images of the scans of fractured spines, and those that are considered normal.

Image Renaming

For easier processing of the image files, each file was renamed using this naming format:

$$\{datasetType\}\_\{classification\}\_\{imageNumber\}$$

Wherein datasetType corresponds to either "train" for training images, and "test" to test images. While classification corresponds to either "frac" for fractured cervix spine images, and "norm" for normal cervix spine images. Finally imageNumber corresponds to the image's number on the collection, which starts from zero(0).

Image Feature Enhancement and Selection

In order to improve the overall performance of the SVM model used, the images undergone feature enhancement and selection using Computer Vision techniques as shown in Figure 2 below.

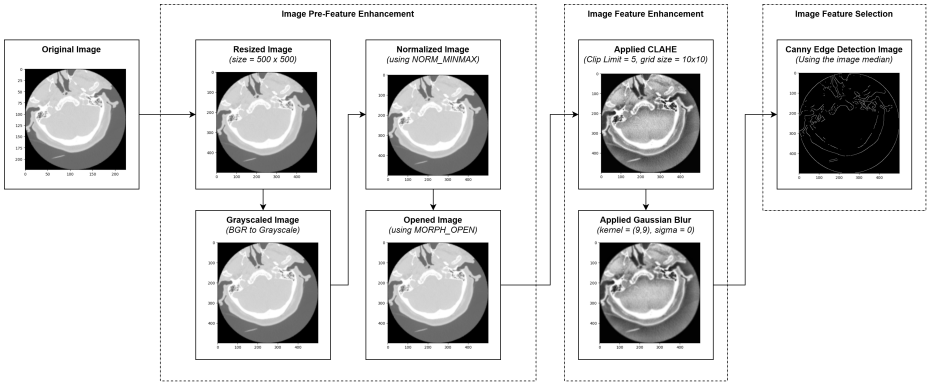

Fig. 2. Overview of Image Feature Selection

From Figure 2, the following Computer Vision techniques and their parameters are used:

1. **Image Pre-Feature Enhancement**

   – *Image Resizing* - All images were resized to 500 pixels by 500 pixels. This was done to ensure that all images in the dataset contains the same dimension. Furthermore resizing the images to a relatively small size ensure a faster and more efficient processing [12].

   This was specifically done using the resize function from open-CV2.

   $$resized\_img = cv.resize(originalimage, (500, 500))$$

   – *BGR Image to Grayscale* - In order to further simplify the processing needed for the images, the team targeted the gray channel of the image by converting the resized image from having three channels (B-Blue, G-Green, and R-Red) to only having one channel (gray).

   This was specifically done using the cvtColor function from open-CV2.

   $$gray\_img = cv.cvtColor(resized\_img, cv.COLOR\_BGR2GRAY)$$

   – *Image Normalization* - Since radiological image scans may vary in color reproduction due to the difference in the scanner specifications that was used, a normalization function was applied to all the images in the dataset. Specifically this was applied to the grayscaled image using the normalize function from open-CV2, with the following parameters:

   i. dst was set to be the same as the grayscaled image.
   ii. alpha = 0, this value served as the lower boundary of the range normalization.
   iii. beta = 255, this value served as the upper boundary of the range normalization.

iv. norm_type = NORM_MINMAX, this meant that the values of each pixels in the image will be normalized from the given alpha-beta range.
v. dtype = CV_8UC1, which corresponds to the color value range of 8-bit unsigned integers from 0 (darkest pixel or black) to 255 (brightest pixel or white).

– *Image Opening* - This technique was utilized to remove the noise from the images. This was specifically done using the morphologyEx function from open-CV2.

$$opened\_img = cv.morphologyEx(norm\_img, cv.MORPH\_OPEN, (1,1))$$

2. **Image Feature Enhancement**
– *Application of CLAHE* - Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to improve the contrast in a given image. This was very useful in enhancing the radiological image, as it emphasizes the bone structure, as well as depth to the radiological image scans of the cervical spine.

This was specifically done by creating a CLAHE object using createCLAHE function from open-CV2 and applying it to the opened image.

$$clahe = cv.createCLAHE(clipLimit = 5.0, tileGridSize = (10,10))$$

$$clahe\_img = clahe.apply(opened\_img)$$

– *Application of Gaussian Blur* - Since the application of CLAHE may also emphasize certain noise in the image, a Gaussian blur was applied to the images to remove such noises.

This was specifically done using the GaussianBlur function from open-CV2 with a kernel size of (9,9), which means that the function takes into account the 9 pixels by 9 pixels grid for the computation of the Gaussian Blur function.

$$blur = cv.GaussianBlur(clahe_img, (9,9), 0))$$

3. **Feature Selection**
– *Canny Edge Detection* - this was utilized to select the median features of the image, which are essentially the edges of the bone structure.
This was specifically done using the Canny function in open-CV2.

$$cannyImg = cv.Canny(blur, np.percentile(blur, 50), np.percentile(blur, 50))$$

2.2 Image Data Pre-Processing

This pre-processing method was utilized primarily to convert the image files to labeled matrices, which can be deployed and reused without affecting the original image files. Whereas the transformed data was saved on a dataframe using pandas, and then the dataframe was pickled using the pickle library from

Python so that it can be utilized in the future for other applications. Specifically, this was done by following the sub-processes indicated in the succeeding subsections. Moreover, it should be noted that the images were already pre-split for both training and testing, as such a method for dataset splitting was not included in this part.

Conversion of Image File to Matrix

The pre-processed images were converted into matrices, as shown in Figure 3, wherein, the shape of the matrix is (500, 500). This means that there are a total of 250,000 values of 0 (black pixels) or 255 (white pixels). However, it must be noted that the Figure 3 does not show the whole matrix as it is too large to fit in the image.

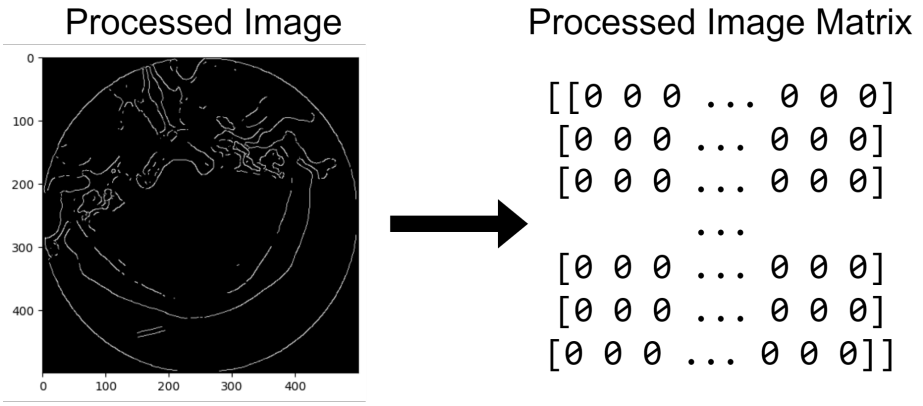

Fig. 3. Visualization of the Conversion Process

Data Labelling and Creation of Data Frames

Corresponding data frames were created for train_normal, train_fractured, test_normal, and test_fractured datasets. Wherein each image matrices was also labelled with "0" for normal, or "1" for fractured.

Pickling of the Data Frames

Using the pickle library in Python, the data frames were pickled, accordingly:

1. df_train.pkl - contains the combined data frames of train_normal, and train_fractured.
2. df_test.pkl - contains the combined data frames of test_normal, and test_fractured.

2.3 Model Training and Evaluation

The data frames of the image datasets have undergone training and evaluation as indicated in the succeeding subsections. Moreover, the training and evaluation were primarily done on a laptop hardware with the following specifications:

- Processor: Intel(R) Core(TM) i5-8300H
- RAM: 16.0GB DDR4 2666Mhz

Meanwhile, the following version of software was used:

- Windows 11 version 22H2
- Python version 3.11.0, with the following additional packages:
  - opencv-python 4.6.0.66
  - matplotlib 3.6.2
  - numpy 1.23.5
  - pandas 1.5.1
  - sklearn 0.0.post1
  - joblib 1.2.0
  - seaborn 0.12.1

About Support Vector Machine (SVM)

Support Vector Machine or SVM, is a machine learning algorithm that aims to identify a hyperplane in a multi-dimensional space, to effectively separate and classify the data points [13]. In this mini-project the group has explored the utilization of three SVM kernels as hyperparameters for the binary classifier for Cervix Spine Fracture. Specifically, the following kernels were utilized:

1. Linear SVM - This is usually used in linearly separable data [14]. Specifically, this was utilized since by observation each pixel in the processed image can only be either black or white.
2. Non-linear SVM
   - Radial Basis Function (rbf) - This is a type of non-linear SVM model, wherein, the similarity between points in the transformed feature space created by the function exponentially decreases [15]. This was utilized in this mini-project as it is commonly used for image classification due to its simplicity and low computational cost.
   - Polynomial (poly) - this is another type of non-linear SVM model which takes additional parameter called a "degree", wherein, a higher degree value results in a more complex model, and at the same time higher computational cost [15]. It was specifically used as part of this mini-project to explore its accuracy in classifying radiological images and how this model would compare to the other two (linear and rbf).

Moreover, it should be noted that the group did not create plots for the decision boundaries for each model as it is physically impossible to plot a decision boundary in 2-D space given the image's dimension of 250,000. And it would not make any sense if the group selected the first two features which are usually only zeros.

SVM Model Training

The image classification model for classifying between normal cervix spine and fractured spine was trained using Support Vector Classifier (SVC) included in Python's sklearn library. Wherein, three SVM model variants was utilized with the following hyperparameter kernels: For Model (a) it utilized a linear kernel; For Model (b) it utilized a radial basis function (rbf); And for Model (c) it utilized a poly kernel. This means that Model (a) utilized a linear SVM, while Model (b) and (c) utilizes a different form of non-linear SVM. Moreover, all models utilized the same value for hyperparameter C (1.0), and hyperparameter gamma was set to default values; except for Model (a), since linear kernel does not need a gamma value [16].

Afterwards each model variants was exported to a compressed pkl file using compression level 5 of the joblib library. This was done so that each model could be deployed outside the notebook, as well as to save time from re-running the training everytime the model is to be used for analysis or evaluation.

Model Evaluation

After the models were trained using the hyperparameters mentioned in the previous section, each model was evaluated using the following:

1. Confusion Matrix - this was used to visualize the different types of errors [17] made by the different models in classifying the given cervix spine X-ray scans.
2. Classification Report Scores [18]
   (a) Accuracy Score - this was utilized to measure the proportion of correct classifications.
   (b) Precision Score - this was utilized to measure the proportion of correct positive classifications.
   (c) Recall Score - this was utilized to measure the proportion of actual positive cases that were correctly classified.
   (d) F1 Score - this was used to get the harmonic mean between precision and recall.
3. Area Under Curve Receiver Operator Characteristic (AUC-ROC) Curve - this was utilized to visualize and better understand threshold between the true positive rate (TPR) and false positive rate (FPR) [19] of the binary SVM-based classifier developed in this mini-project.

3 Results and Discussions

3.1 Confusion Matrices

The confusion matrix in this mini-project was used to illustrate the performance of the SVM classifier models on the testing samples with known actual

classes. This was used to analyze and determine the correctly and incorrectly classified samples which were essential to determine the efficiency of the models developed.

Confusion Matrix for Model (a)

Figure 4 shows that Model (a) has correctly classified 355 samples out of 400. More specifically, 180 over 200 fractured bones are classified as a fracture (true positive), and 175 over 200 normal bones are not classified as a fracture (true negative). On the other hand, the model has incorrectly classified 45 samples out of 400. It consists of 20 over 200 fractured bones classified as normal and 25 over 200 normal bones classified as fractured.

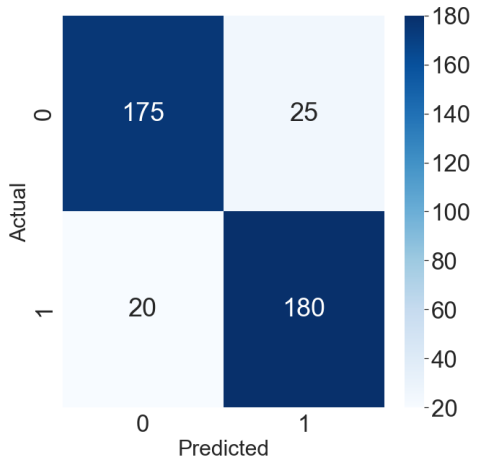

Fig. 4. Confusion Matrix for Model (a)

Confusion Matrix for Model (b)

Figure 5 shows that Model (b) has correctly classified 387 samples out of 400. More specifically, 189 over 200 fractured bones are classified as a fracture (true positive), and 198 over 200 normal bones are not classified as a fracture (true negative). On the other hand, the model has incorrectly classified 13 samples out of 400. It consists of 11 over 200 fractured bones classified as normal and 2 over 200 normal bones classified as fractured.

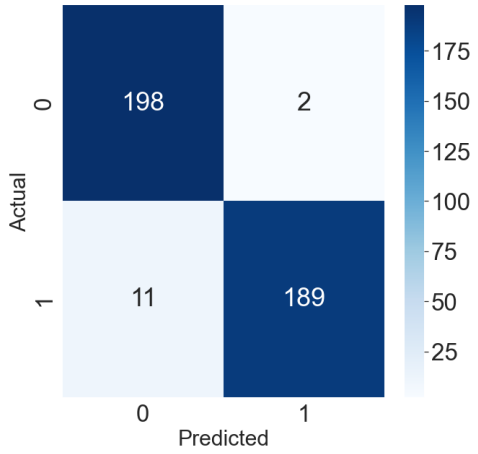

Fig. 5. Confusion Matrix for Model (b)

Confusion Matrix for Model (c)

Figure 6 shows that Model (c) has correctly classified 204 samples out of 400. More specifically, all 200 fractured bones are classified as a fracture, however, only 4 over 200 normal bones are not classified as a fracture. On the other hand, the model has incorrectly classified 196 samples out of 400 which are solely normal bones classified as fractured.

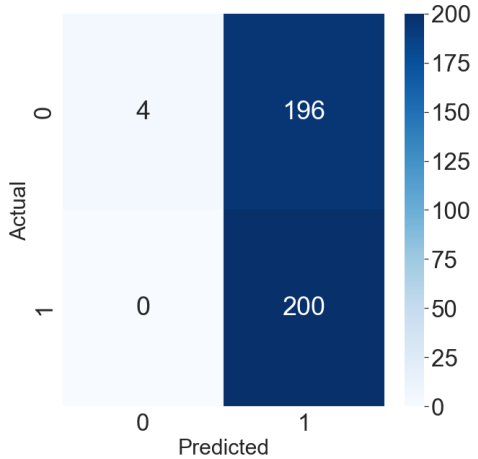

Fig. 6. Confusion Matrix for Model (c)

The classification result showed that Model (b) has the most numbers of correctly classified samples which are 189 true positives and 198 true negatives. On the other hand, Model (c) has the least number of correctly classified samples with only 4 true negatives despite of a perfect number of true positives.

3.2 Accuracy, Precision, Recall, and F1 Scores

The classification report in this mini-project shows the accuracy, precision, recall, and F1 scores of the three SVM models. Accuracy was used in this project to calculate the number of correctly classified classes. Moreover, precision was used to calculate how many of the predicted fractured bones are actually fractured, while recall was used to calculate how many of the actual fractured bones are predicted as fractured. Lastly, the F1 score is used to compute the harmonic mean of the precision and recall values.

Accuracy

Table 1 shows that Model (b) has the highest likelihood of correctly classifying samples with an accuracy of 0.9675 or 96.75%, followed by Model (a) with an accuracy of 0.8875 or 88.75%, and lastly by Model (c) with an accuracy of only 0.5100 or 51%. Furthermore, the accuracy of Model (c) implies that almost half of its predictions are incorrect.

Table 1. Accuracy of the three models

Model	Accuracy
(a)	0.8875
(b)	0.9675
(c)	0.5100

Precision

Table 2 shows that Model (b) has the highest likelihood of having predicted fractured bones that are actually fractured with a precision of 0.9895 or 98.95%, followed by Model (a) with a precision of 0.8780 or 87.80%, and lastly by Model (c) with a precision of only 0.5000 or 50%. Moreover, the precision of Model (c) implies that only half of the bones that it classified as fractured are actually fractured bones.

Table 2. Precision of the three models

Model	Precision
(a)	0.8780
(b)	0.9895
(c)	0.5000

Recall

Table 3 shows that Model (c) has the highest likelihood of having actual fractured bones that are predicted as fractured with a precision of 1.0000 or 100%, followed by Model (b) with a recall of 0.9450 or 94.50%, and lastly by Model (a) with a recall of 0.9000 or 90%. However, it should be noted that Model (c) gained 100% recall due to the fact that it labelled almost all training dataset images as fractured cervical spine images, as such if we take a look back at Table 2, we can see that its precision is only at 50%. With this we can say that Model (b) still performs better by the rules of precision/recall tradeoff [20].

Table 3. Recall of the three models

Model	Recall
(a)	0.9000
(b)	0.9450
(c)	1.0000

F1 score

Table 4 shows that Model (b) has the highest likelihood of having predicted fractured bones that are actually fractured and having actual fractured bones that are predicted as fractured. It has an F1 score of 0.9668 or 96.68%, followed by Model (a) with an F1 score of 0.8889 or 88.89%, and lastly, by Model (c) with an F1 score of 0.6711 or 67.11%. This results for the F1 scores is strongly beneficial in understanding the precision/recall tradeoffs for all three models. Wherein, it was shown that the F1 score of Model (c) suffered despite it having a 100% recall, due to its very low precision score of 50%.

Table 4. F1-scores of the three models

Model	F1-score
(a)	0.8889
(b)	0.9668
(c)	0.6711

Summary of Scores

As shown in Table 5, Model (b) outperformed the other models in most measures of classification and was closely followed by Model (a). On the other hand, Model (c) showed relatively low scores with the exemption of recall wherein the model achieved 100%. Even so, it is worth noting that Model (c)'s perfect recall is only due to the model classifying all images as fractured images, thus, all positive samples are automatically detected. This resulted in a very high number of false negatives hence the low accuracy and precision. Furthermore, this classification report shows that the polynomial kernel performs relatively poorly compared to the other kernel functions (*i.e.,* linear and rbf). It is possible that this was caused by the kernel's default parameters not being optimized for the images used as samples.

Table 5. Summary of Scores for the three models

Model	Accuracy	Precision	Recall	F1-Score
(a)	0.8875	0.8780	0.9000	0.8889
(b)	0.9675	0.9895	0.9450	0.9668
(c)	0.5100	0.5000	1.0000	0.6711

3.3 AUC-ROC Graph

The Receiver Operator Characteristic (ROC) curve was used in this mini-project to plot the true positive rate (TPR) against the false positive rate (FPR) across different threshold values. The Area Under the Curve (AUC) was then used to summarize the ROC curve and measure the models' ability to correctly identify classes. Essentially, a classification model with a high (i.e., close to 1) AUC value is more capable of distinguishing between classes.

Figure 7 compares the ROC curve and AUC of the three models. It shows that Model (b)'s curve is closest to the top-left corner of the ROC graph and therefore, has the highest AUC which is 0.9978 or 99.78%. This denotes that among the three models, Model (b) has the highest sensitivity and specificity which corresponds to a higher likelihood of correctly classifying all positive and negative class points. Furthermore, Model (b) was followed by Model (a) with an AUC of 0.9597 or 95.07% and lastly by Model (c) with an AUC of 0.8911 or 89.11%.

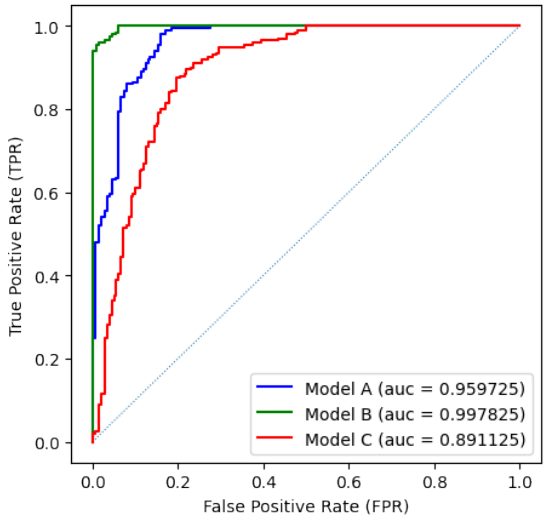

Fig. 7. AUC-ROC Graph for the Three Models

3.4 Run-time Comparison Across the Three Models

Shown in the next table are the run-times for each of the models' predictions of 400 test image data. Model (a) was the fastest with 135.45 seconds, Model (c) came in second with 227.39 seconds, and finally Model (b), with 1,536.62 seconds. In other words, Model (a) was able to classify around 3 cervical spine X-ray images per second. While Model (b) and (c) was able to classify around 0.26 and 2 images per second, respectively. Moreover, with this, despite Model (b) performing better than Model (a) in the previous classification Metric Report, the speed at which Model (a) can classify a cervical fracture can be proven beneficial, especially for emergency situations as suggested by the Radiological Society of North America [21].

Table 6. Summary of Run-times for the three models in seconds

Model	Run-time (in seconds)
(a)	135.45
(b)	1536.62
(c)	227.39

4 Conclusions

In this mini-project, the team has successfully developed a binary classifier for Cervix Spine Fracture Detection using Enhance Canny Edge Detection as the primary feature selection technique and Support Vector Machine as the primary machine learning algorithm for classifying between normal and fractured cervical spine. Specifically, this mini-project presented the following key findings:

1. The combination of Computer Vision techniques to enhance the radiological image scans proved favorable in enhancing the bony structures of the radiological image scans, as suggested by the high-performance results of the SVM models in classifying between normal and fractured cervical spine radiological images.

   More specifically, selecting the median gray-scale value as a parameter for canny edge detection shows to be a great value to acquire at least 90% accuracy.

   Furthermore, this also suggests that bone structures for any radiological scans may be detected using its median value alone, which could be used to improve other existing models for other radiological scans, which requires examination of the bone structure. This technique is also supported by the paper of Hwa, Bade, and Hijazi for COVID-19 and Pneumonia X-ray Images [8].

2. The confusion matrices showed that Model (b) has the highest true positive and true negative predictions, by correctly classifying 189 out of 200 fractured cervical spine, and 198 out of 200 normal cervical spine, respectively. This was closely followed by Model (a), and Model (c) performed the worst. However, this is probably due to the fact that the default parameters used for Model (c) is way beyond the optimal parameters needed to classify the given images.

3. SVM model (b) is the highest-performing model based on its accuracy, precision, recall, and f1 scores.

   This is closely followed by the SVM Model (a). While SVM Model (c) is the worst-performing SVM model variant explored in this mini-project, as shown previously in Table 5.

   Moreover, this shows that radial basis function (rbf) and linear are highly favorable SVM parameters to be used in radiological image classification.

4. The AUC-ROC curve also showed that Model (b) has the highest sensitivity and specificity with AUC percentage value of 99.78%, which means that out of the three models it has the higher likelihood of classifying all positive and negative class points. This is again closely followed by Model (a) with

AUC percentage of 95.97%. While Model (c) is the lowest with the AUC percentage of 89.11%.

5. SVM Model (a) runs the fastest compared to the other two models, followed by SVM Model (c) and lastly by SVM Model (b) as shown previously in Table 6.
   The trade between accuracy and run-times were visible between Model A's shorter run-time but slightly lower accuracy against Model B's drastically longer run-time but with a very promising accuracy.

   Moreover, this shows that utilizing SVM with linear hyperparameters could be highly favorable to be used in situations where the speed of the classification would matter more than its accuracy, for example in emergency situations. It is also important to keep in mind that these predictions' run-times were timed after finishing 400 predictions of processed image data; therefore, in real-life scenarios, the said models can likely perform much faster if given a smaller amount of image data to process.

4.1 Recommendations

Due to the limitations in the conduct of this mini-project, there are some things that the team would have done better, as such they are presenting the following recommendations for anyone who would like to continue or improve on this project:

1. Pre-processing parameters used could be adjusted to further explore its effects on the scores of the SVM model
2. Finding the optimal hyperparameter values for C and gamma for the given SVM models using Grid Search. This was specifically explored by the team during the conduct of this mini-project, however, due to hardware limitations, the process was not able to be completed in a reasonable time; and
3. The developed classifiers, specifically Model (a) and (b) may be deployed into a web or mobile application for the utilization of medical professionals.

References

1. "Cervical fracture (broken neck)." [Online]. Available: https://orthoinfo.aaos.org/en/diseases--conditions/cervical-fracture-broken-neck/
2. "Cervical spine fractures & dislocations," Dec 2019. [Online]. Available: https://www.uscspine.com/conditions-treated/neck-disorders/cervical-spine-fractures-dislocations/
3. M. D. Ross and J. M. Cheeks, "Undetected Hangman's Fracture in a Patient Referred for Physical Therapy for the Treatment of Neck Pain Following Trauma," *Physical Therapy*, vol. 88, no. 1, pp. 98–104, 01 2008. [Online]. Available: https://doi.org/10.2522/ptj.20070033
4. "Neurosurgeon: Neck fractures can go undetected initially," Jan 2012. [Online]. Available: https://www.cbsnews.com/pittsburgh/news/neurosurgeon-neck-fractures-can-go-undetected-initially/
5. R. Mammadli, "What are cervical spine fracture complications?" Aug 2016. [Online]. Available: https://iytmed.com/cervical-spine-fracture-complications/
6. Radiological Society of North America, "Rsna cervical spine fracture ai challenge (2022)." [Online]. Available: https://www.rsna.org/education/ai-resources-and-training/ai-image-challenge/cervical-spine-fractures-ai-detection-challenge-2022
7. I. Kumar, J. Rawat, and H. Bhadauria, "A conventional study of edge detection technique in digital image processing," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 4, pp. 328–334, 2014.
8. S. K. T. Hwa, A. Bade, and M. H. A. Hijazi, "Enhanced canny edge detection for covid-19 and pneumonia X-Ray images," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 979, no. 1, p. 012016, Nov. 2020.
9. L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," *Michigan State Universiy*, vol. 2, no. 2, p. 4, 2006.
10. C. Li, S. Zhang, H. Zhang, L. Pang, K. Lam, C. Hui, and S. Zhang, "Using the k-nearest neighbor algorithm for the classification of lymph node metastasis in gastric cancer," *Computational and mathematical methods in medicine*, vol. 2012, 2012.
11. I. Nitze, U. Schulthess, and H. Asche, "Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification," *Proceedings of the 4th GEOBIA, Rio de Janeiro, Brazil*, vol. 79, p. 3540, 2012.
12. R. Asokan, "Basics of computer vision: 1. interpolation and resizing," 2021. [Online]. Available: https://raghul-719.medium.com/basics-of-computer-vision-1-image-resizing-97fca504cd63
13. R. Gandhi, "https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47," 2018.
14. Java T Point, "Support vector machine algorithm." [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
15. Geeks for Geeks, "Introduction to support vector machines (svm)," 2022. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/
16. S. Yıldırım, "Svm hyperparameters explained with visualizations," 2020. [Online]. Available: https://towardsdatascience.com/svm-hyperparameters-explained-with-visualizations-143e48cb701b

17. Indeed Editorial Team, "What is a confusion matrix? (with definition and steps)," 2022. [Online]. Available: https://ca.indeed.com/career-advice/career-development/confusion-matrix
18. R. Banerjee, "Understanding accuracy, recall, precision, f1 scores, and confusion matrices," 2021. [Online]. Available: https://towardsdatascience.com/understanding-accuracy-recall-precision-f1-scores-and-confusion-matrices-561e0f5e328c
19. i2 Tutorials, "Auc-roc curve- it's importance in machine learning," 2020. [Online]. Available: https://www.i2tutorials.com/auc-roc-curve-its-importance-in-machine-learning/
20. G. Aurelien, *Chapter 3: Classification.* O'Reilly, 2020.
21. Radiological Society of North America, "Ai helps radiologists detect bone fractures," 2022. [Online]. Available: https://www.sciencedaily.com/releases/2022/03/220329114710.htm