

# Bone Fracture Classification

By: David Solow, Eshan Bhatnagar, Kristen Cirincione, and Brian Xiao

## Abstract

The bone fracture classification dataset consists of a collection of X-ray images capturing different types of bone breaks. The images cover a range of ten different bone fracture classes. The images in this dataset are not restricted to a specific region of the body. There is also significant variation in the size of the images. For instance, some images are as small as 77x125, whereas others are as large as 640x640. Additionally, the orientation of each image is not standardized. Finally, the images also exhibit varying intensities. Some X-ray images appear to have a very high contrast between the bone and background while others appear more muted.

The intent of this project is to classify each X-ray image into one of these ten bone fracture categories using logistic regression, support-vector machine, and random forest classifiers with the original images and the following engineered features: Histogram of Gradients, Canny Edges, Contours, VGG19 Transfer Learning, and Principal Component Analysis of pixel data.






These model and feature combinations only achieved a 43% accuracy on the test set at best. It was noted that the training accuracy achieved 100% which indicates poor generalization and overfitting. In future research we hope to augment our dataset with complete manual labeling of fracture bounding boxes, an approach we partially experimented with in this research.






## 1. Introduction

A fracture is the medical term used to describe a break in the continuity of a bone. There are a multitude of different fracture types, each varying in severity. Due to the variations in severity, it is imperative medical professionals be able to identify the type of fracture before developing the most appropriate treatment plan. Not all fracture types are easy to identify, therefore, we believe it would be of great value for doctors to have access to a model that can assist in this classification task.

## 2. Dataset

The dataset used throughout the entirety of this paper can be found at the following location: [bone fracture dataset](#)<sup>1</sup>. This dataset consists of a total of 1129 JPEG X-Ray images spanning ten different fracture types. Table 1 provides a description of each of the ten fracture types along with an example image and the number of images in the dataset associated with that specific fracture type.

Fracture Type	Description	Example	Number of Images
Avulsion fracture	Occurs when a fragment of bone is pulled off by a tendon or ligament. It typically happens in areas where a tendon or ligament attaches to the bone.		125
Comminuted fracture	A fracture where the bone is broken into several pieces. This type of fracture is often the result of high-impact trauma.		150
Fracture Dislocation	Involves both a fracture and a dislocation of a joint. The bone is not only broken but also displaced from its normal alignment.		158
Greenstick fracture	A fracture where the bone bends and cracks without breaking completely. It is most commonly seen in children.		124
Hairline Fracture	A small crack or severe bruise within a bone. It is also known as a stress fracture and is often caused by overuse or repetitive force.		113

Impacted fracture	A fracture where the broken ends of the bone are driven into each other. This often occurs in falls or car accidents.		86
Longitudinal fracture	A fracture that runs along the length of the bone. It is usually caused by direct trauma.		82
Oblique fracture	A fracture that occurs at an angle across the bone. It is typically caused by a sharp blow or indirect force.		87
Pathological fracture	A fracture that occurs in a bone weakened by disease, such as osteoporosis, cancer, or infection. It happens with minimal trauma or stress.		136
Spiral fracture	A fracture where at least one part of the bone has been twisted, resulting in a spiral-shaped break. It is often caused by a twisting force.		88

*Table 1. Bone Fracture Classes*

Based on the information provided in both Table 1 and Figure 1, you can see that our dataset is slightly imbalanced. There are several more instances of images showing fracture dislocations, comminuted, and pathological fractures (158, 150, and 136 images respectively) than images showing impacted, longitudinal, oblique, and spiral fractures (86, 82, 87, and 88 images respectively). Whether this discrepancy in the number of image samples is large enough to meaningfully impact the results of our models will be discussed in the Models section of this paper.

In addition to the variation in the number of images per fracture class, there also exists variation in the size of the images and the contrast of the images. Figure 2 displays the differences in image dimensionality across our dataset. As you can see, there is quite a broad range of image sizes, with the smallest being 77x125 and the largest being 640x640. A majority of the images appear to trend towards larger dimensions, with a mean image height of 560 pixels and a mean image width of 430 pixels. There is also significant variation in the contrast of our images. Figure 3 shows the distribution of the root mean square contrast of all the images across our dataset. The RMS contrast of the images appears to be somewhat normally distributed with a mean value of approximately 0.27. Both the variations in size and contrast will be addressed in the preprocessing of our images before model development.

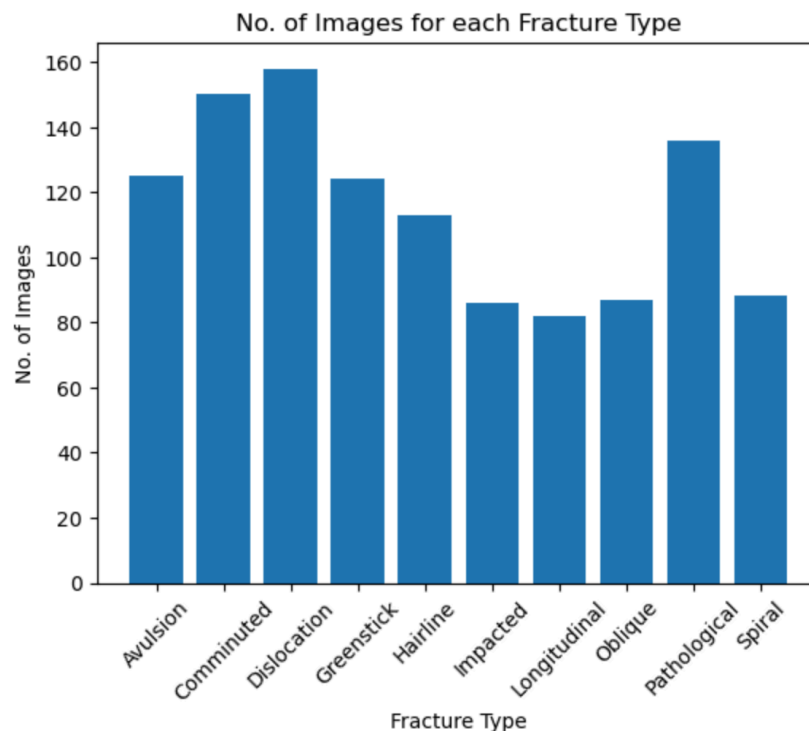


Figure 1. Number of Images in the dataset associated with each fracture type

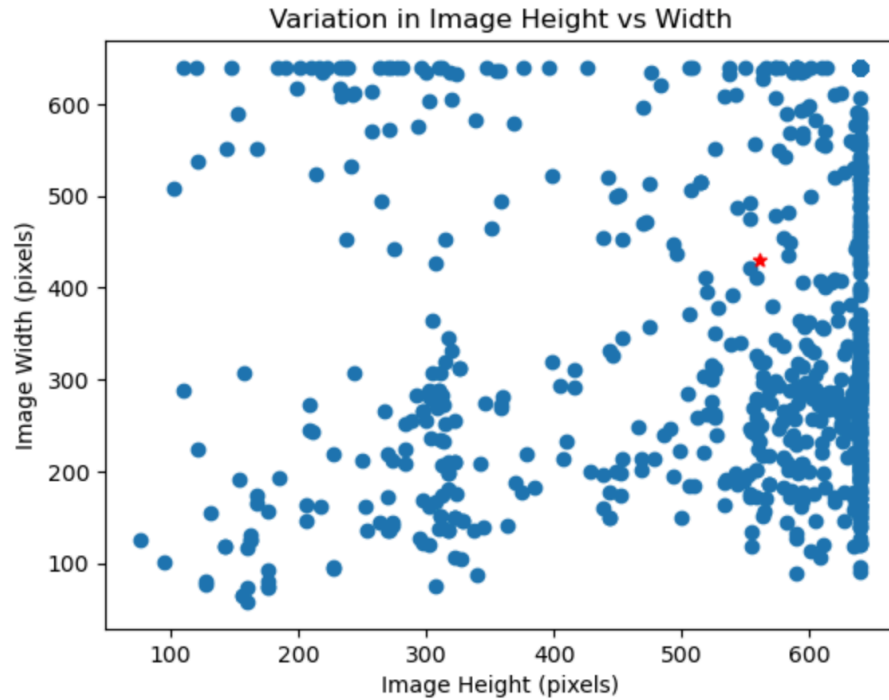


Figure 2. The height and width of each image in the dataset in pixels. The star represents the mean image height and width.

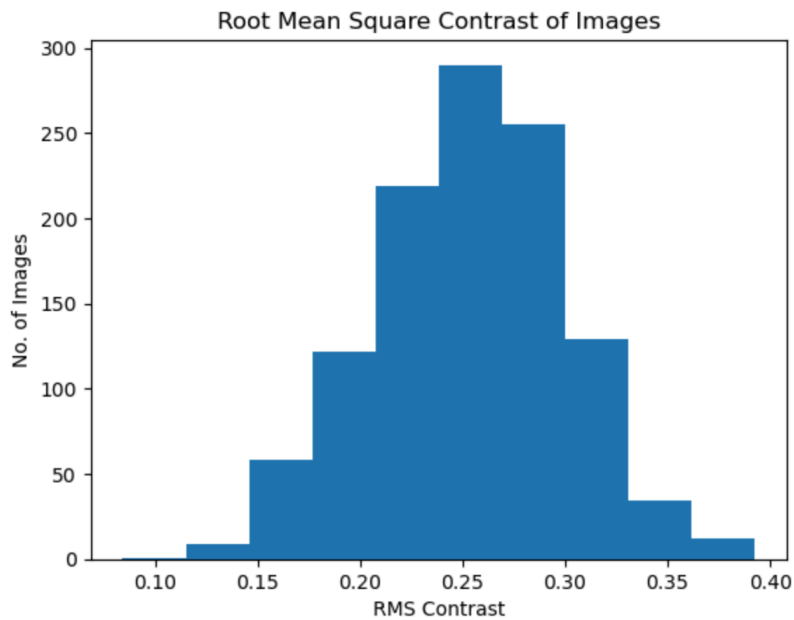


Figure 3. Root Mean Square (RMS) contrast of dataset images

It is also worth explicitly noting that the images in our dataset are not limited to a specific region of the body (as is demonstrated in Table 1). Although the images can capture a fracture in any location in the body, this is not to say that certain fracture types are not partial to specific regions

of the body. For example, greenstick fractures often occur in the arms of children, therefore, it follows that most of our X-Ray images of greenstick fractures capture the arm.

### **3. Preprocessing**

Prior to feature extraction, a collection of preprocessing techniques were applied to each image in our dataset. First, each image was converted from RGB to grayscale, allowing us to reduce the complexity of our image data from three color channels to a single intensity channel. Given the black and white nature of X-Rays, this reduction in complexity allows our models to converge faster without adversely affecting the classification of the fractures.

In addition to grayscale conversion, we also performed histogram equalization on all of our images. Histogram equalization is a means of contrast adjustment. We chose to apply it to our images to mitigate the variations in contrast shown in Figure 3. Once applied, histogram equalization standardized the RMS contrast of almost all of our images to approximately 0.5.

Next, in preparation for model training, we chose to standardize the dimensions of our images. In order to balance the costs of downsampling with the costs of training models on larger images, we chose to resize our images to 512x512 pixels. At this size we were able to slightly reduce the computational cost of training our models without jeopardizing the clarity of smaller fractures.

Lastly, all pixel values were scaled to values between 0 and 1. This allows models with gradient descent based algorithms to converge more quickly. We also experimented with unsharp masking, but found that it did not improve the final accuracy of our models, thus we omitted it from our final image preprocessing pipeline.

Data loading, preprocessing, and feature extraction were packaged into a command line interface tool to assist with recreation of our data and models by other researchers.

## **4. Feature Extraction**

### **4.1 Simple Features**

Three simple features were selected to draw out more information from the fracture images and ultimately assist in building our classification models. These features included Histogram of Oriented Gradients (HOG), Canny edges, and contours, all of which are shown in Figure 4 for a single X-Ray image of a comminuted fracture.

HOG captures the distribution of edges oriented in different directions within an image and computes their gradients. This is particularly useful for fracture identification as HOG can recognize the shape and structure of the bone, which can inevitably be altered by fractures. HOG can also capture changes in patterns of the bone's shape by focusing on edge directions.

Additionally, it can illuminate the bone's shape through the intensity of the gradients, which is useful for images that vary in lighting quality.

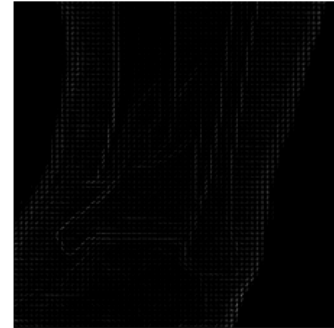
The Canny edge detector is similar to HOG in its ability to identify edges in an image by detecting areas of rapid intensity change. These edges make it useful for identifying discontinuities in bone structure<sup>2</sup>. The detector's noise reduction helps minimize the effect of image noise, and the algorithm is also able to identify the boundaries of fractures by detecting the locations of the edges.

Contours identify areas in the image that have the same intensity, making it useful in capturing the shape of the bone structures as well as highlighting the areas where the normal shape may be disrupted by a fracture.

Original Image: Comminuted Fracture



HOG Image



Canny Edges



Contours



*Figure 4. Illustration of of HOG, Canny Edges, and Contours on a Comminuted Fracture*

As a means of further exploring our simple features, we conducted principal component analysis (PCA) to reduce the dimensionality on all feature vectors. This created a visual representation of the fraction of the total variance explained by the number of principal components as shown in Figure 5.

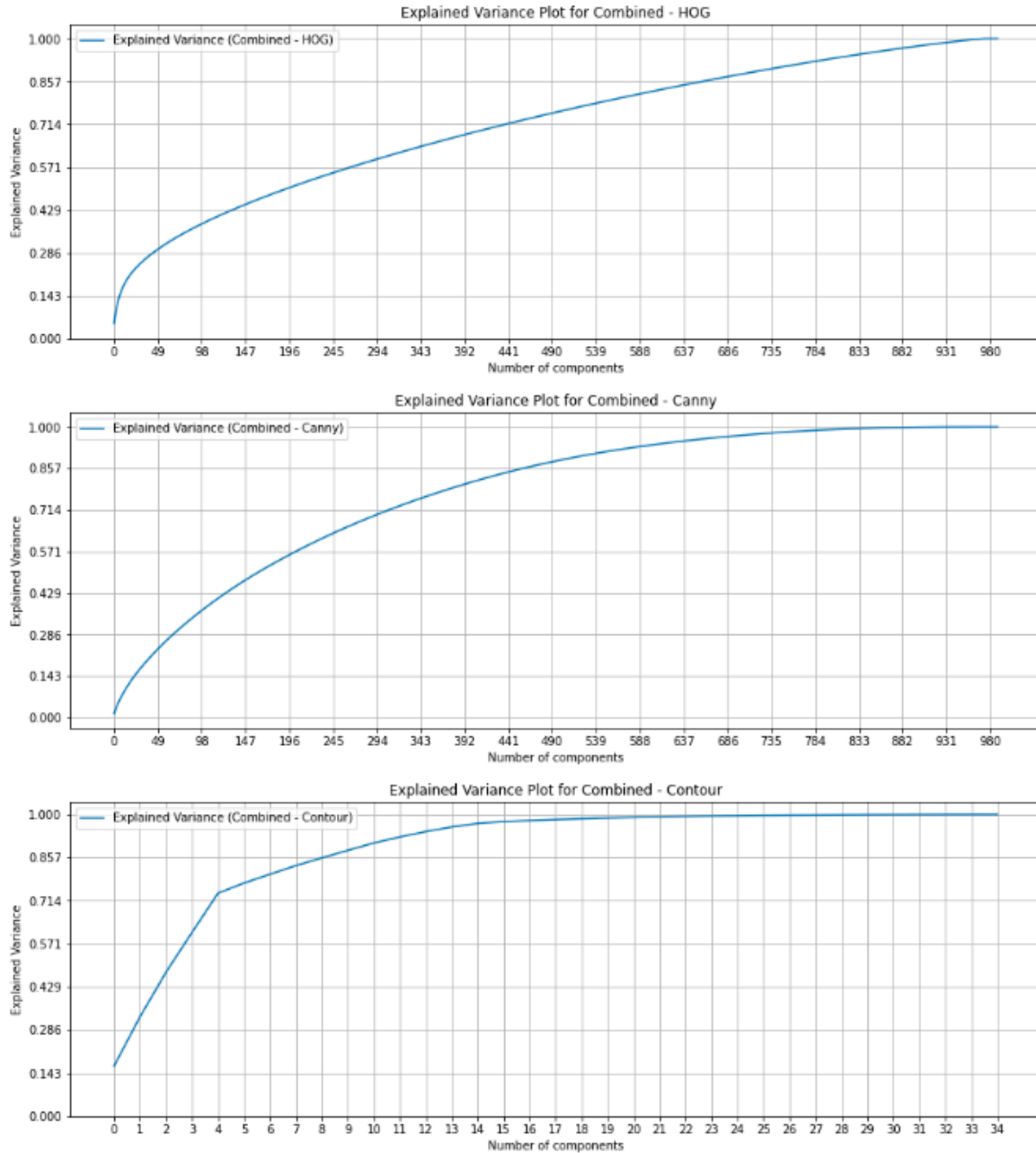
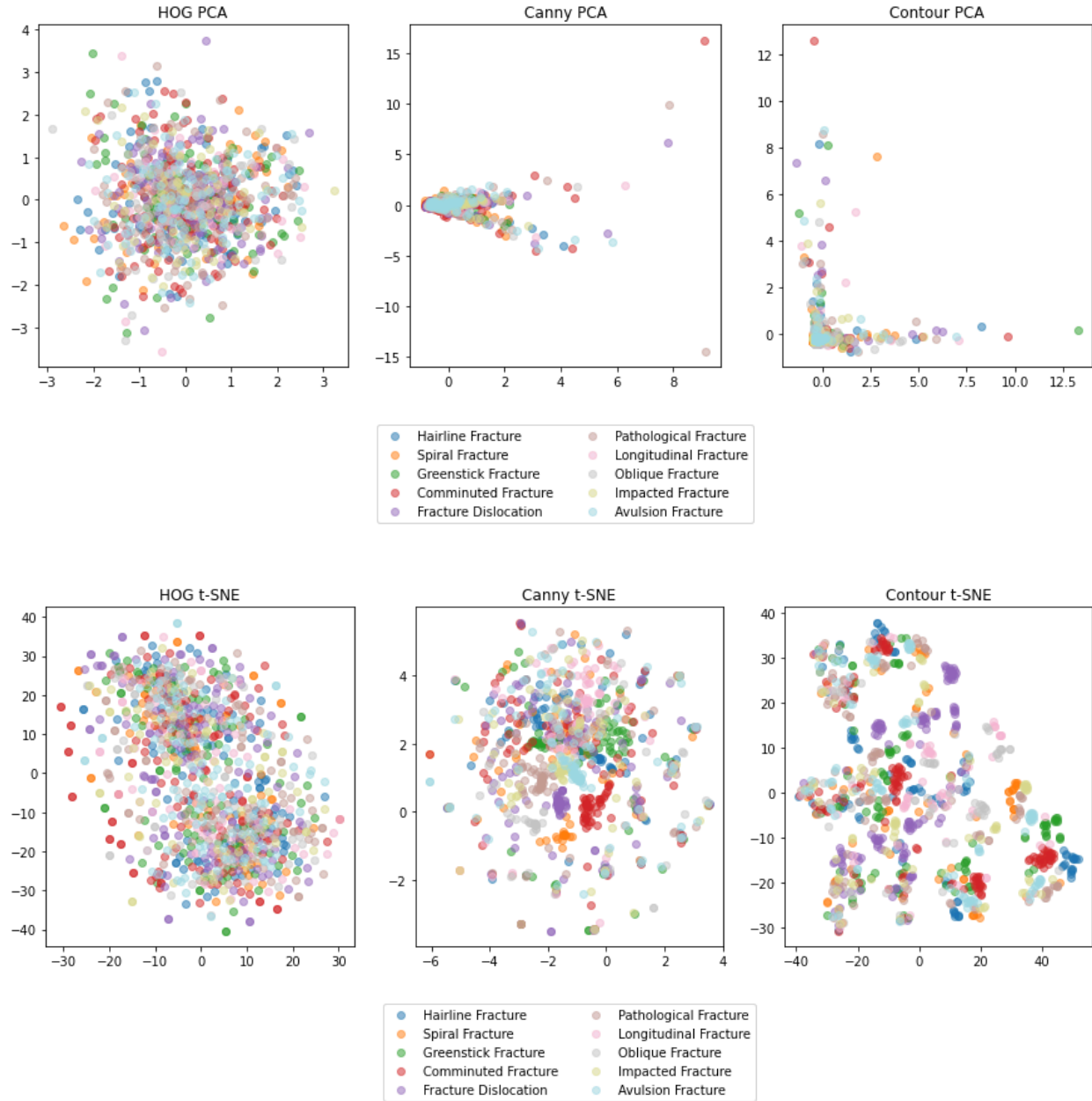


Figure 5. Explained variance plot from principal components of HOG, Canny edge, and contour features

Looking at the graphs for contours and Canny edges in Figure 5, we can see that about half of the principal components account for nearly 90% of the variance. On the other hand, the HOG graph has a slower and more gradual increase, indicating that more principal components are needed to explain the variance, and thus the higher dimensionality of this feature is more significant.





*Figure 6. PCA and t-SNE visualizations of first two components on the simple features of the training dataset*

Looking at the plots for the PCA on the three simple features in Figure 6, it can be seen that most of the variance is distributed along the first two principal components for the Canny edges and contours features. T-distributed Stochastic Neighbor Embedding (t-SNE) was also used as a dimensionality reduction technique to help visualize the clusters of fracture classes for each feature. We can see some of those clusters beginning to form for the contours features, however, these clusters are nowhere near as distinct in the HOG and Canny edges features.

PCA was also applied on the raw images and used as an additional simple feature to see if it would improve model performance compared to just using the images themselves.

## 4.2 Complex Feature

Other researchers have had much success when it comes to identifying bone fractures using transfer learning techniques<sup>3</sup>, therefore, we chose to extract features from the VGG-19 model. The VGG-19 model consists of a convolutional neural network (CNN) that is 19 layers deep and is pre-trained on the ImageNet database. We ran each of our images through this model and extracted the final layer to use as features in our fracture classification model. Each X-Ray image ended up with an associated feature array consisting of 16x16x512 elements. Figure 7 visualizes 6 of these 16x16 features for a single spiral fracture X-Ray. Although not immediately apparent, these features ended up being quite useful in our fracture classification models.

Original Image: Spiral Fracture



VGG-19 Features

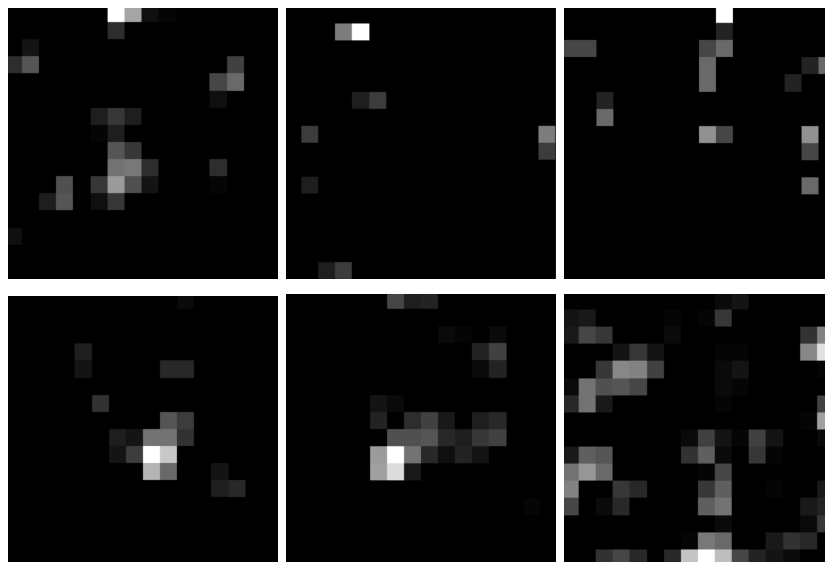


Figure 7. VGG-19 features visualized for a spiral fracture

## 5. Models

### 5.1 Test, Train, Validation Split

The data for each image class was already split into train and test sets, with the train set including 989 images and the test set containing 140 images. After preprocessing all the images and extracting the features, the train vectors for each feature and the baseline images were passed to each of our models (Logistic Regression, Support Vector Machine, Random Forest). The models were run once on each of the feature vectors (HOG, Canny edges, contours, PCA of the raw images, VGG-19) and the baseline image vectors. An additional run was conducted combining the baseline images with the best performing feature (VGG-19). However, the model just trained on the VGG-19 features ended up performing better than all other feature combinations for all three classifiers. Therefore, a hyperparameter search was used to further optimize the performance of this model. The hyperparameter search method we chose (RandomizedSearchCV) implements a cross validation technique that splits the training data into folds. The hyperparameter search iterates through the folds, using one fold as the validation set and the remaining folds as the training set until all folds were used once as a validation set. This methodology allows the hyperparameter search algorithm to select hyperparameters that achieve the best performance both in terms of accuracy and generalizability. The best hyperparameters for each of our three models are shown in Table 2. These values were used in our models when predicting on our test set of data.

Model	Parameter	Search Values	Best Value
Logistic Regression	max_iter	100, 500, 1000, 2000	500
	C	0.01, 0.1, 1, 10, 100	1
	multi_class	'multinomial', 'ovr'	'ovr'
Support Vector Machine	max_iter	500, 1000, 2000, 10000	1000
	C	0.1, 1, 10, 100	100
	kernel	'linear', 'rbf', 'poly'	'linear'
	gamma	'scale', 'auto'	'scale'
Random Forest	n_estimators	100, 1000, 10000	1000
	criterion	'gini', 'entropy'	'entropy'
	max_features	'sqrt', 'log2'	'sqrt'
	bootstrap	True, False	False

*Table 2. Hyperparameter Tuning*

For logistic regression, the parameters chosen included 'max\_iter', which corresponds to the maximum number of iterations taken for the solvers to converge. C was used as the inverse of the regularization strength, where a smaller value specifies stronger regularization. Since logistic regression is often used for binary classification, and this is a multiclass problem, we wanted the hyperparameter search to determine if it would be best to use a multinomial strategy or a 'One versus Rest' strategy. In 'One versus Rest', a binary classification is used for each label, where the positive class is the current class and the negative class is a grouping of all the other classes. On the other hand, 'multinomial' is typically used for multiclass problems but uses a multinomial cross entropy loss approach. Ultimately, our hyperparameter search determined the 'One versus Rest' strategy would be best.

For the support vector machine, maximum iterations was once again used as a parameter, and C was once again used as a regularization parameter, acting as the inverse of the regularization strength. Another parameter was the kernel used to transform the input data to a higher dimensional space. Linear acts as the simplest kernel, as it just calculates the dot product for two vectors, just like a linear classifier, instead of transforming the data. The Radial Basis Function (RBF) is a Gaussian kernel that maps the data into an infinite-dimensional space and is useful when the relationship between the labels and attributes is nonlinear. The polynomial kernel is similar to RBF but is useful if there are polynomial relationships between the variables. In addition to the kernels, 'gamma' acts as a kernel coefficient parameter where 'scale' sets the value to  $1 / (\text{number of input features} * \text{variance of input features})$  and 'auto' simply sets the value to  $1 / (\text{number of features})$ .

For the random forest model, the 'n\_estimators' parameter refers to the number of decision trees used in the forest. The 'criterion' parameter measures the quality of a split at each node in the trees, with 'gini' measuring how often a randomly chosen element is improperly classified, and 'entropy' measuring how much information is gained by splitting the node. The 'max\_features' parameter controls the number of features to consider for each node split, using either the square root or the logarithm (base 2) of the number of features. Setting the bootstrap value to true or false determines whether each tree is trained on a random subset of the data with replacement (which could lead to more diverse trees) or with the entire dataset (which could reduce model variance but increase overfitting).

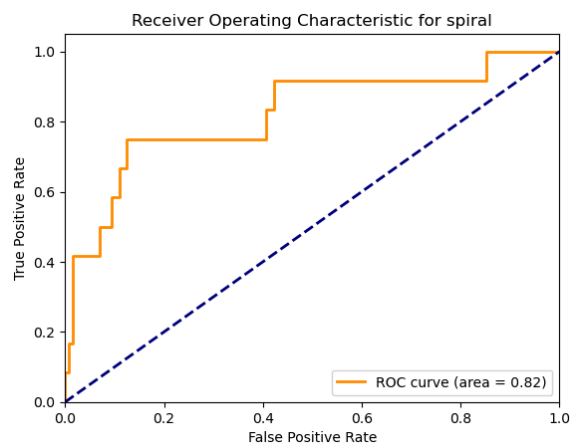
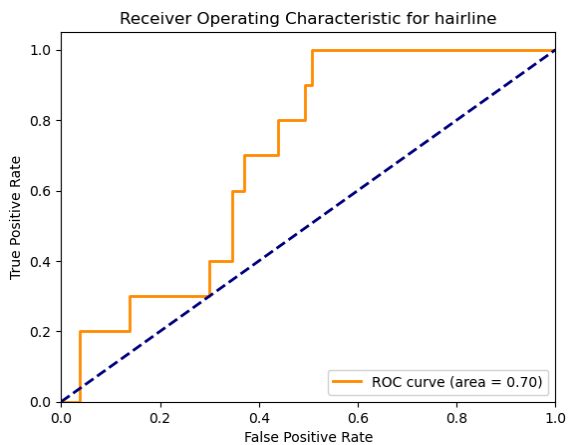
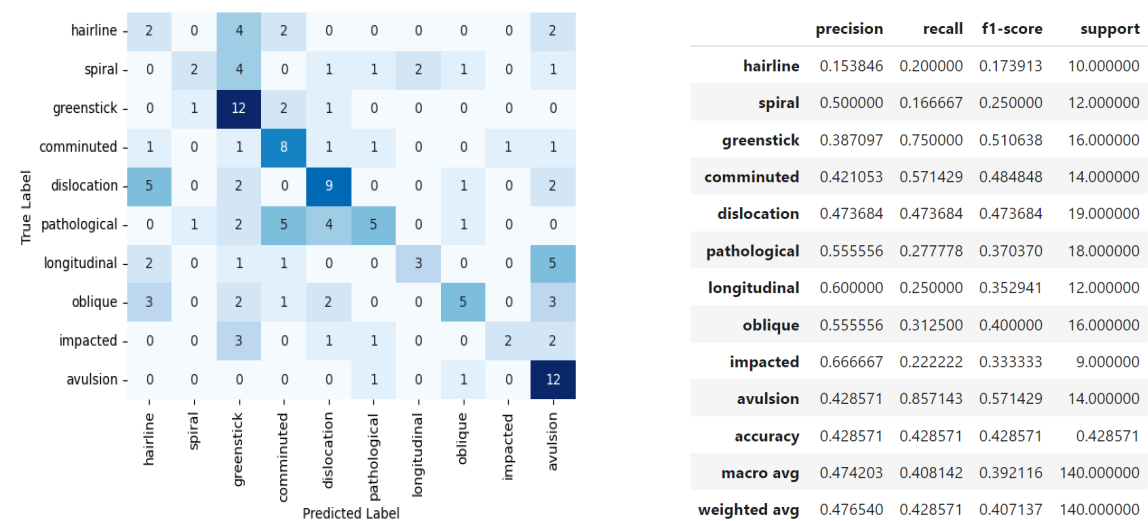
## 5.2 Logistic Regression Model

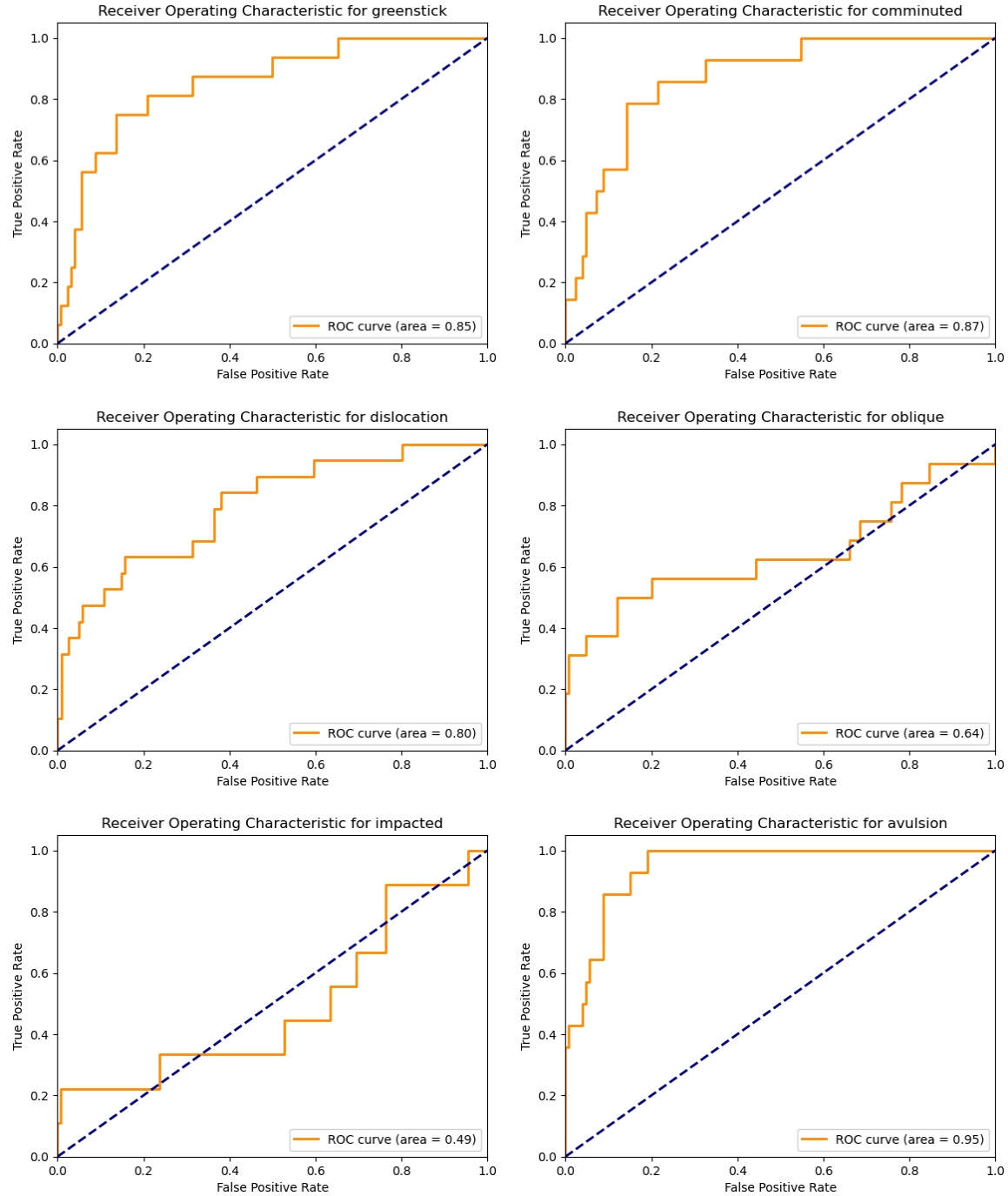
Logistic Regression performed the best on the test set out of all three models used, achieving the highest overall accuracy of 42.86% and recall of 40.81%. It was also the fastest model to be trained and had the quickest inference time, needing only 773 seconds and 0.71 seconds respectively. The best performing classes were: greenstick, avulsion, comminuted, and dislocation, which all achieved an AUC of at least 0.8. However, only the recall for greenstick, avulsion and comminuted fractures were above 50%. We believe that our model performed the best on these fracture classes due to their highly distinctive nature relative to the other fracture classes. For instance, it is not particularly easy to distinguish between a hairline and a longitudinal fracture due to their similarity in appearance, however, it is quite easy to distinguish

between a comminuted (bone broken in multiple places) and a hairline fracture. Greenstick, comminuted, and avulsion fractures are simply more visually distinctive, thus easier to classify.

Logistic Regression is the simplest model which makes it less prone to overfitting when compared to more complex models like Random Forest. This combination of simplicity and best performance on the test set means it generalizes better than our other models. This model is limited in the sense that it assumes the features have a linear relationship with the outcomes, which may not be true especially for high dimensional problems like image processing. To improve upon this model, we can incorporate L1 or L2 regularization to further prevent overfitting and improve the accuracy in the test set.

The results of this classifier are shown below in Figure 8, with the confusion matrix shown first followed by the ROC curves for each class.





*Figure 8. Results of the Logistic Regression Classifier using VGG19 only. In the first row are the confusion matrix, and table of results for each class. The remaining rows are the ROC curves with AUC calculated for each class.*

### 5.3 Support Vector Machine (SVM) Model

SVM performed the second best on the test set out of all three models used, achieving an overall accuracy of 40.71% and recall of 39.41%. It was the second fastest model to be trained and produce an inference, needing 3714 seconds and 13.5 seconds respectively. This is about

3000 seconds longer for training and 12 seconds longer for inference when compared to logistic regression. The best performing classes are the same as the logistic regression model: greenstick, avulsion, comminuted, and dislocation, which all achieve an AUC of at least 0.8. However, only the recall for greenstick, avulsion and comminuted fractures are above 50%.

SVM aims to find the hyperplane that maximizes the margin between classes, which can lead to better generalization, especially when classes are well-separated. However, it is clear that fractures are not always easily identifiable, with some fractures appearing to be very similar to one another. This may be why SVM performs slightly worse than logistic regression, even though both are linear classifiers.

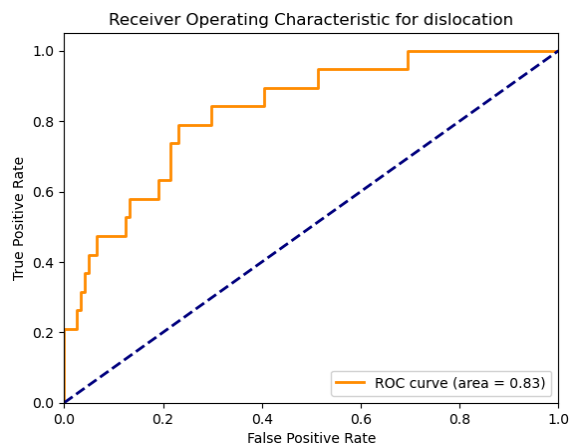
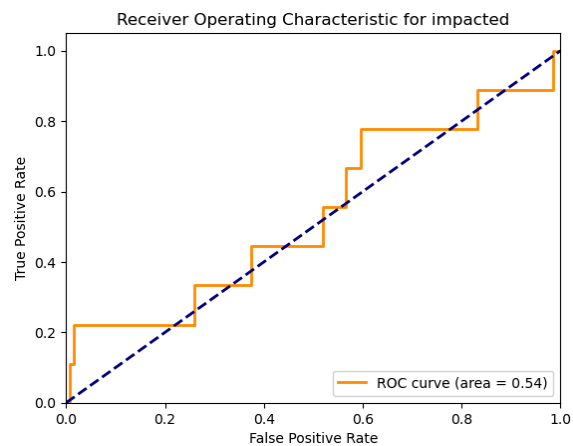
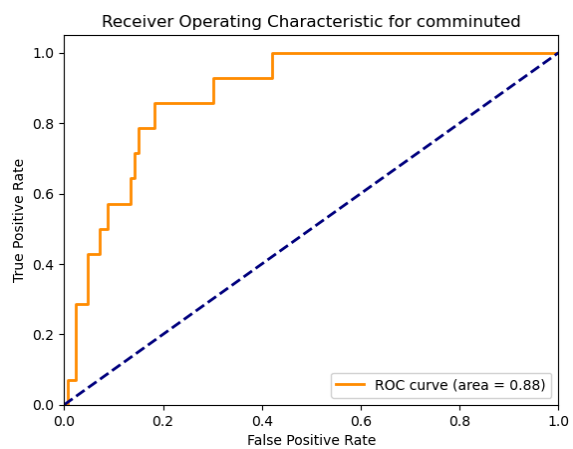
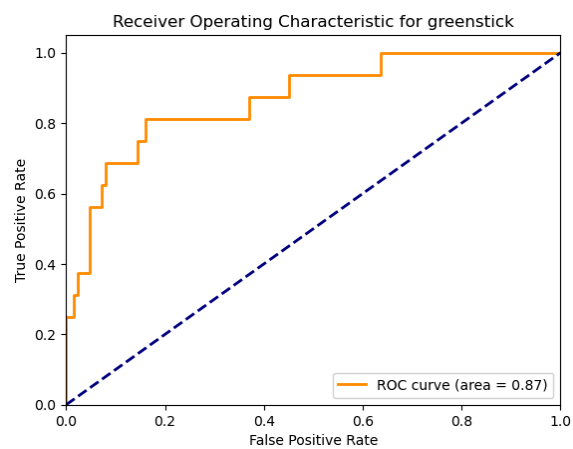
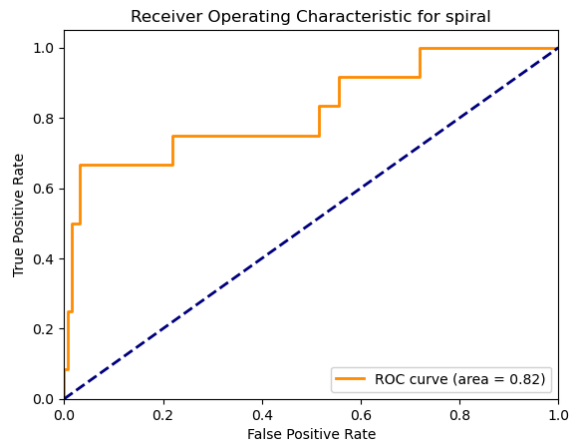
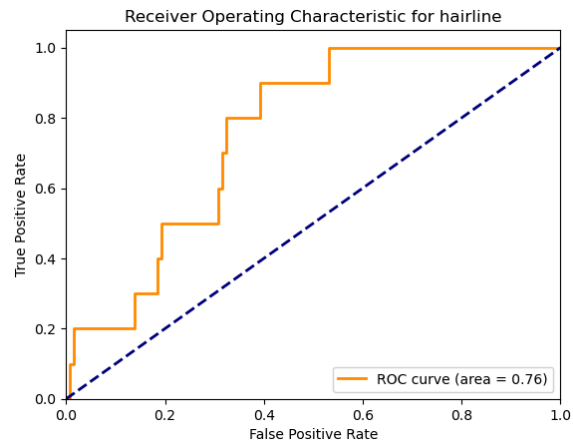
The limitations of this model are similar to that of logistic regression in that it assumes the features have a linear relationship with the outcomes, which may not be true for high dimensional problems like image processing.

To improve upon this model, we can continue to experiment with the regularization parameter, C, with the goal of creating a more generalizable model.

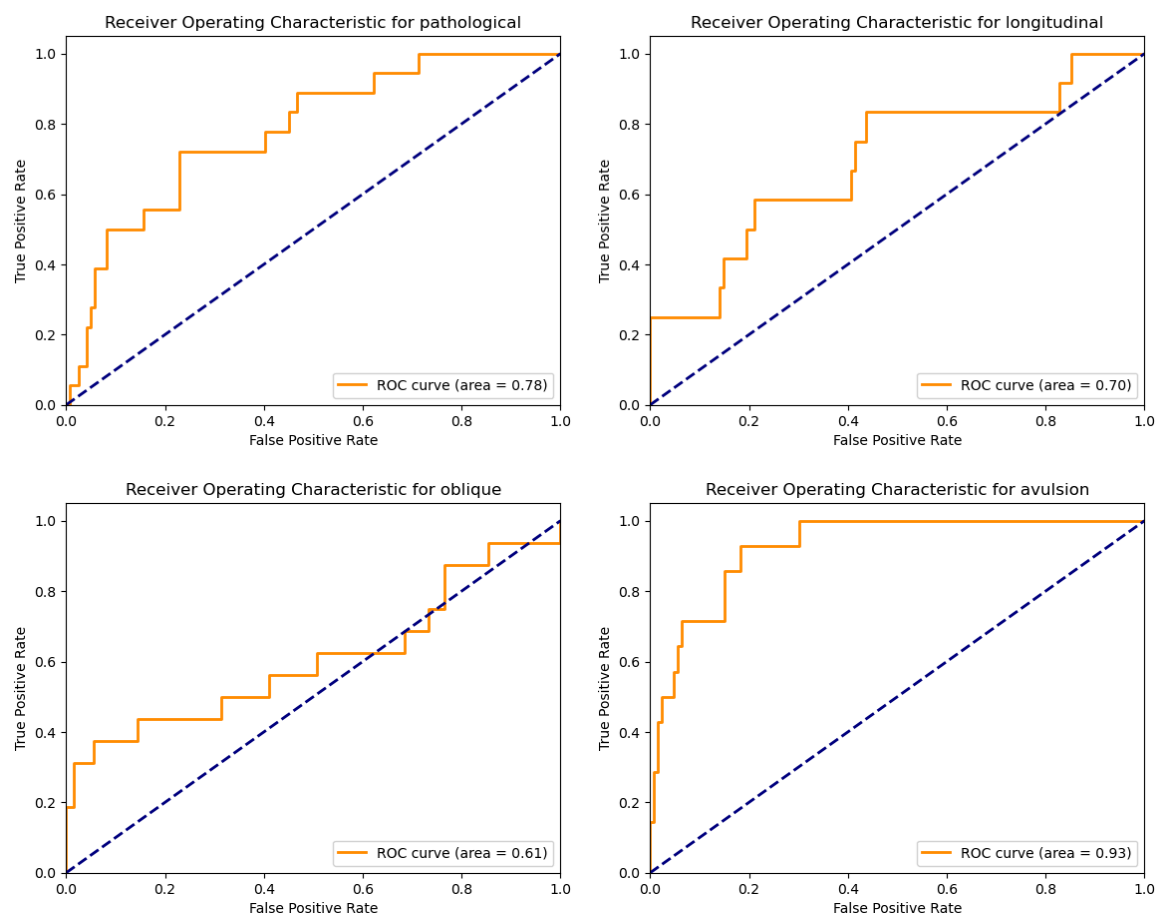
The results of this classifier are shown below in Figure 9, with the confusion matrix shown first followed by the ROC curves for each class.

hairline	2	0	4	2	0	0	0	0	0	2
spiral	0	3	1	2	1	1	2	1	0	1
greenstick	1	1	11	2	1	0	0	0	0	0
comminuted	1	0	1	9	2	1	0	0	0	0
dislocation	5	0	2	0	7	1	0	1	2	1
pathological	0	0	2	5	4	4	0	1	1	1
longitudinal	2	0	1	0	1	0	3	0	0	5
oblique	3	0	2	1	1	1	0	5	0	3
impacted	0	0	3	0	1	1	0	0	2	2
avulsion	1	0	0	2	0	0	0	0	0	11
	hairline	spiral	greenstick	comminuted	dislocation	pathological	longitudinal	oblique	impacted	avulsion

	precision	recall	f1-score	support
hairline	0.133333	0.200000	0.160000	10.000000
spiral	0.750000	0.250000	0.375000	12.000000
greenstick	0.407407	0.687500	0.511628	16.000000
comminuted	0.391304	0.642857	0.486486	14.000000
dislocation	0.388889	0.368421	0.378378	19.000000
pathological	0.444444	0.222222	0.296296	18.000000
longitudinal	0.600000	0.250000	0.352941	12.000000
oblique	0.625000	0.312500	0.416667	16.000000
impacted	0.400000	0.222222	0.285714	9.000000
avulsion	0.423077	0.785714	0.550000	14.000000
accuracy	0.407143	0.407143	0.407143	0.407143
macro avg	0.456346	0.394144	0.381311	140.000000
weighted avg	0.460301	0.407143	0.391377	140.000000







*Figure 9. Results of the SVM Classifier using VGG19 only. In the first row are the confusion matrix, and table of results for each class. The remaining rows are the ROC curves with AUC calculated for each class.*

## 5.4 Random Forest Model

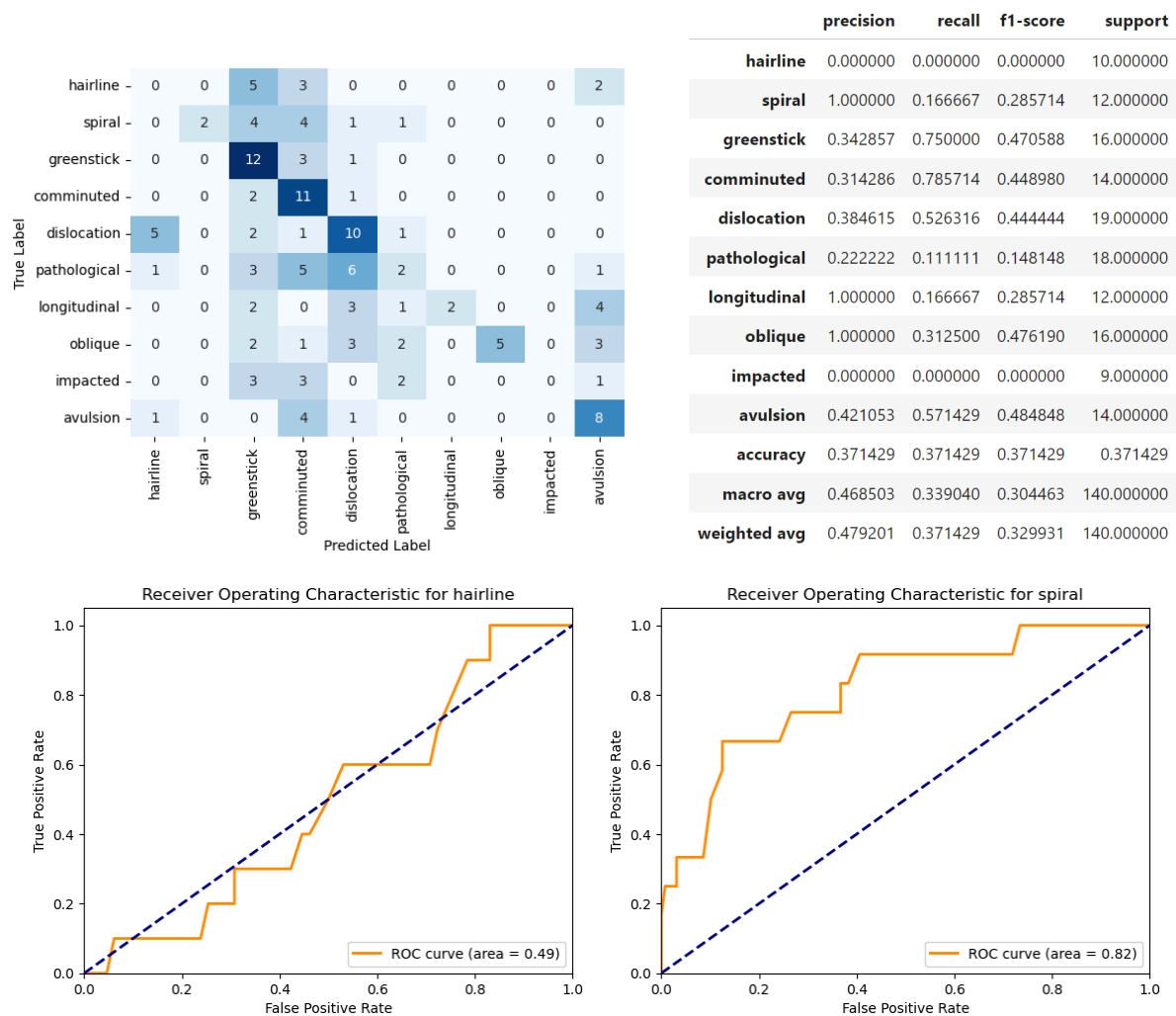
Random Forest performed the worst on the test set out of all three models used, achieving an overall accuracy of 37.14% and recall of 33.90%. It was by far the slowest model to be trained but second fastest when it came to inference, needing 9149 seconds and 9 seconds respectively. This is about an additional 5500 seconds needed to train when compared to the SVM, but about 4 seconds faster to inference. The best performing classes are the same as the Logistic Regression and SVM models: greenstick, avulsion, comminuted, and dislocation, which all achieve an AUC of at least 0.8. However, only the recall for greenstick, avulsion and comminuted fractures are above 50%.

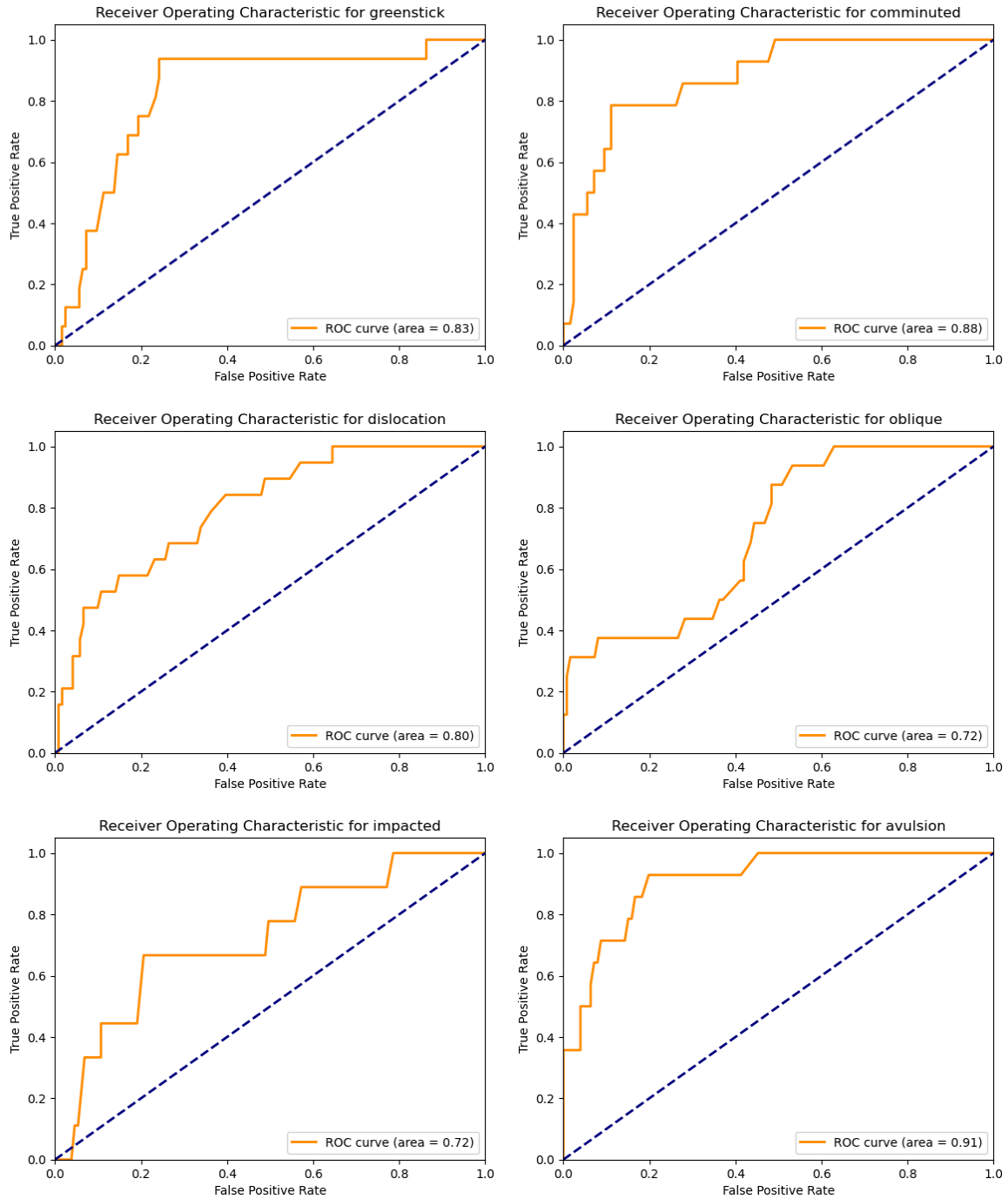
Random Forest is the most complex model when compared to Logistic Regression and SVM. It also introduces non-linearities which may be the cause of the long training time. This model also appears to be overfitting to the training set and thus generalizing poorly.

This model is limited in the sense that it is more prone to overfitting and can be more difficult to tune due to the increase in hyperparameters. Additionally, Random Forest models often require more data than linear classifiers.

To improve upon this model, we can add more parameters to the gridsearch such as max\_depth, min\_samples\_split, and min\_samples\_leaf. However, including these parameters will significantly increase the gridsearch time.

The results of this classifier are shown below in Figure 10, with the confusion matrix shown first followed by the ROC curves for each class.





*Figure 10. Results of the Random Forest Classifier using VGG19 only. In the first row are the confusion matrix, and table of results for each class. The remaining rows are the ROC curves with AUC calculated for each class.*

## 5.5 Model Efficiency vs Model Accuracy

Metric	Logistic Regression	SVM	Random Forest
Accuracy	42.86%	40.71%	37.14%
Training Time	773 seconds	3714 seconds	9149 seconds
Inference Time	0.71 seconds	13.50 seconds	9.43 seconds

*Table 3. Model Results and Metrics*

According to the results of all models shown in Table 3, the Logistic Regression model is both the most accurate and most efficient model making it an easy choice as the best model.

For the medical space, accuracy should be the highest priority as these decisions will impact patient's health directly. Any misclassifications can have severe negative consequences including loss of life or financial repercussions for the hospital. It must be noted that some fractures are extremely hard to detect even by medical professionals which may contribute to the poor accuracy. Hairline fractures and pathological fractures for example, are known to be difficult fractures to detect.

Training time is most likely the least important metric for these models. Fractures are not an evolving space, where they change or look different (simply put, an avulsion fracture is an avulsion fracture). The model does not need to be constantly re-trained to keep up with new classes.

Inference time is very important, as these models need to have a prediction generated by the time the radiologist reviews the x-rays. Depending on the workflow and severity, this could be a few minutes so the inference time should aim to be as fast as possible. For all our models, the inference time is well under a minute, which means we may have room for more advanced models such as CNNs which may improve upon the accuracy as well.

## 6. Conclusion

While the maximum performance achieved by our models and features may not be sufficient for clinical use, we believe that the performance achieved indicates the feasibility of performing fracture classification using even simple linear models with the correct preprocessing and feature extraction. Furthermore, the limited size of our dataset combined with the significant dropoff in performance between train and test sets indicates that the poor test performance observed may be due to overfitting issues caused by our limited data both in aggregate and per class. Despite this, our results indicate the potential of the features we've used, in particular the use of transfer learning features from pre-trained models.

In future research, we hope to augment this dataset using both simple alteration techniques like rotation and flipping as well as the expansion of our preliminary tests involving the manual

labeling of bounding boxes around fracture regions, which we hope to replace with complete labeling of the entire dataset by qualified physicians. We believe that these interventions will help resolve challenges with overfitting and help the model generalize sufficiently for viable use in a clinical setting. Finally, in the future we hope to compare our feature-extraction based approach with state of the art CNN and visual transformer architectures to understand if similar performance can be achieved with simpler models and fewer parameters.

## References

- [1] Darabi, P.K., "Bone Break Classification Image Dataset," Kaggle, 2021. Available at: <https://www.kaggle.com>
- [2] Muhammet Emin Sahin, Pongsakorn Samothai, D. P. Yadav, Basha, and Rao, L.J., "Bone Fracture Classification Using Transfer Learning," arXiv preprint arXiv:2406.15958, 2024. Available at: <https://arxiv.labs.arxiv.org/html/2406.15958>
- [3] Chitkara University School of Engineering and Technology, Hekma School of Engineering, Computing and Informatics, University of Turku, and University of Monastir, "Hybrid SFNet Model for Bone Fracture Detection and Classification Using ML/DL," Sensors, vol. 22, no. 15, p. 5823, 2022. doi:10.3390/s22155823. Available at: <https://www.mdpi.com/1424-8220/22/15/5823>