Group 3 : ADS2002 Project Report

Advancing Catheter Placement Accuracy: A Comprehensive Pipeline for Detection and Malposition Analysis

Name	Monash ID
Gideon Cher	33154333
Terence Tan	33973156
Jiazhe Han	33072493
Joshua Hudson	33890269
Mehak Dhingra	32484712
Tyasha Perara	32474148
Simin Liang	33499179

Executive Summary 2
Introduction 5
Model Development and Results 8
Conclusion 19

Group 3

Due Sunday, 20 October 2024 11:55 PM

MONASH University

Executive Summary

Known to be an integral part of contemporary healthcare, enabling the precise administration of fluids, medications and treatments, Catheters, when positioned accurately are essential for stabilizing patients and promoting recovery, particularly within intensive care environments. However, the ramifications of mispositioned catheters can be severe, leading to a series of much greater harm than initially intended. These misaligned tubes can lead to serious complications, including organ damage, respiratory distress, infections and even death. Thus, it is vital for all medical facilities to have the best resources to eradicate any error and diminish future damage from incorrect placement. Given the frequent overcapacity of hospitals – exacerbated by the COVID – 19 pandemic – the automation of catheter placement identification has become imperative for enhancing patient safety. Thus, this project aims to address this critical challenge through the application of many machine learning methodologies to develop results which are capable of discerning the presence of both proper positioned and mispositioned catheters, maintained through a range of Convolutional Neural Networks (CNN's) and similar data science tools.

The dataset used in this project consists of a large collection of chest radiographs with annotations. This data is from a publicly available dataset on Kaggle, specifically designed for catheter line classification tasks. The catheter dataset includes multiple variables, such as:

- Patient ID (e.g.
- Catheter brand (e.g.)
- Positioning status (normal, borderline or abnormal).

1) **Normal** refers to when the catheter is correctly positioned and functioning as intended.
2) **Borderline** refers to when the catheter may need slight repositioning but is still functioning adequately.
3) **Abnormal** positioning refers to when the catheter is incorrectly positioned and requires immediate correction to avoid serious complications.

Each radiograph is annotated to indicate whether a catheter is present and whether or not it is correctly positioned. The dataset provides a detailed analysis of catheter placement in different scenarios and patient profiles, allowing us to explore the factors that affect the accuracy of placement status.

Thus, taking advantage of machine learning techniques such as the broadness of neural networks and their unique offerings, we aimed to construct models capable of distinguishing the nearness of catheters, anticipating malpositioned ones and generating annotations directly on medical images in a competent manner. Utilizing the RANZCR-Clip dataset, which contains thousands of labelled datasets, our focus centred on three different research questions. Intuitively speaking, a project of this sophistication which consists of such complex concepts requires great attention to detail and a broad grasp on the hypotheses of the project, therefore, rather than

mistakenly beginning the project with a side of ignorance, research proved more than relevant and complement the grand successes of the project as a whole. Accordingly, the research questions were prompted as follows with each further question being a basis of the first:

- Firstly, can we predict correct and incorrect catheter placement at a high standard?
- Furthermore, can we determine the brand and position of catheters using image classification models?
- Can a model take an input radiograph and output precise annotations of catheter locations?
- Which specific catheter types are more prone to misplacement.

These questions were all explored throughout the project and helped gain a deeper understanding of the results established. This catheter project followed a specific methodology that involved three major phases, each targeted a specific aspect of catheter management using unique computer models.

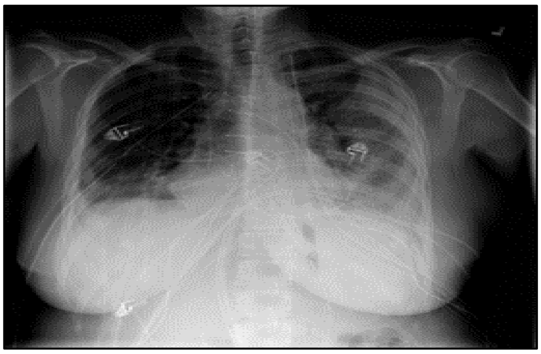


Figure 1: A visualisation of what each image entailed. Showed after grey scaling each image.

Employing:

- **YOLOv8** for binary classification, we were attempting to conclude on the hypothesis whether we could predict the correctness or incorrectness of a catheter's placement. The choice of the specific neural network YOLO8 to resolve this question was proposed by an article which exposed the successful results it acquired when using this model to improve female breast cancer, titled "Exploring the potential of...... YOLOv8 in improving breast cancer awareness". Thus, a new manipulated dataset gained after we re-defined what was considered normal and abnormal labelling catheters based on their correctness. Furthermore, splitting this newly manipulated dataset into training (70%), test (20%), validate (10%), we further used three different pixel depths on our Yolov8 model, being 256 x 256, 512 x 512 and 1024 x 1024, with the help from a resizing software, Power Toys. Thus, using all three of Yolov8's models, (small, medium and large over CSPDarknet53) **we concluded with model results successfully detecting 80% of false negatives, underscoring its capability to assist clinicians by minimizing undetected malpositioned catheters correctly.**

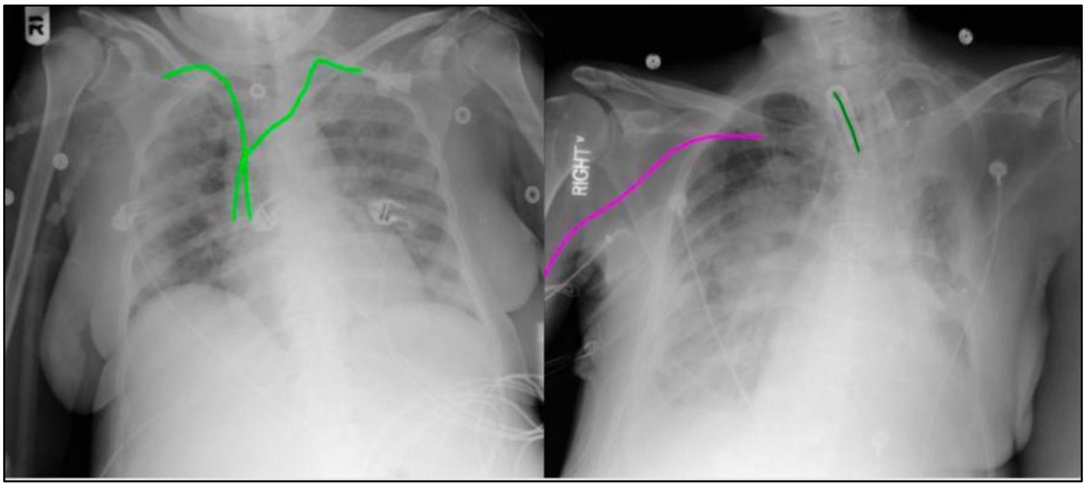


Figure 2: A visualisation example of two images which have their annotations of their catheters outlined by colour. Each colour is a different brand. E.g. we see that the right image has two brands of catheter.

- **ResNet and DenseNet** multilabel classification models for identifying the brand and position of catheters these models were found to be rather successful due to their superior performance on image classification tasks. These models usually excel at recognising complex patterns such as catheter types and positions, even in images which consist of more noise such as rescaled images. **The results however show ResNet achieving an AUC score of 87%, reflecting a rather pleasing result. However, with an even greater performance, DenseNet reluctantly even outperformed ResNet with an AUC score of 91%, demonstrating superior capability for multi – label classification.**
- **UNet** for annotation and segmentation, the CNN's (convolutional neural network) segmented the inputted image and predicted the location of each catheter, outputting a guide of annotations to guide clinicians in assessing the placement. However, this was done with a slightly modified dataset of images. This newly acquired dataset consisted of images overlayed with their corresponding catheters highlighted, this helped acquire the results for this model. **This specific UNet model achieved 94% validation accuracy, accurately detecting catheter placement and improving high quality segmentation maps.**

Despite the grand set of successful results which were the results of our modelling endeavours, there were several changes which could've slightly further enhanced our results. The computational limitations played a rather large detriment on the limited timing of the project as when handling such large detests, running some of these neural networks with specific GPU's can lead to occasional delays and errors. Yet, despite these delays and small setbacks, the models were able to demonstrate the potential of machine learning techniques to revolutionize the medical industry by increasing patient safety regarding catheter placement. With models such as YOLOv8, ResNet, DenseNet and UNet, clinicians can receive faster and more accurate assessments of catheter placements, minimizing human error and preventing complications.

Introduction

At hospitals and medical facilities, one of the most important lifesaving medical equipment are catheters. Catheters are tubes placed into your body which allow the body to have fluids delivered to the bloodstream, administer medications, allow for urine to drain freely, or to assist in various other medical procedures or surgeries (Brown, 2018). One of the main places that a catheter can be placed is within a patients' chest, this type of catheter is what we will explore. In recent years the need for catheters within hospitals has exponentially increased, with COVID-19 greatly increasing the demand, as well as people experiencing chest pain or coronary artery disease all requiring catheters placed in the chest. Correct placement of these catheters is extremely crucial in order to maintain effective treatment and patient safety. Catheters that are malpositioned or have been inserted incorrectly can lead to various complications, such as organ injury and even death. Catheter placement is a game of mm, as an incorrectly placed catheter can injure veins, cause internal bleeding, puncture your lungs, or affect your heart's rhythm (Brown, 2018). Complications with catheters can also occur outside the body, but for this report we will investigate possible ways to mitigate the chance of errors of internal catheter placement.

The dataset used in this inquiry consists of a large collection of chest radiographs with annotations. This data is from a publicly available dataset on Kaggle, specifically designed for catheter line classification tasks and exploratory analysis. The dataset was created by 'Royal Australian & NZ College of Radiologists', in a bid to help find a way to identify incorrect catheter placements through the use of neural networks made by the public and data science community (Royal Australian & NZ College of Radiologists, 2024).The catheter dataset includes multiple variables, such as patient ID, catheter brand, and positioning status (normal, borderline or abnormal). Normal refers to when the catheter is correctly positioned and functioning as intended. Borderline refers to when the catheter may need slight repositioning but is still functioning adequately. Abnormal positioning refers to when the catheter is incorrectly positioned and requires immediate correction to avoid serious complications. Each radiograph is annotated to indicate whether a catheter is present. The dataset includes several types of catheters, each serving a different medical purpose. The Endotracheal Tube (ETT) is used to assist breathing by providing an airway, with the tip of the catheter only having a 2 cm margin of error. An incorrect placement of this can lead to the collapsing of the lungs. The Nasogastric Tube (NGT) is inserted through the nose into the stomach for feeding or draining, with its placement being greater than 10 cm away from the gastro-oesophageal junction. The complication of misplacing this is that it can be inserted into the lungs instead of creating a hole in the oesophagus. The Central Venous Catheter (CVC) is designed to deliver medications or fluids directly into large veins, and the Swan Ganz Catheter is used to measure pressures in the heart and arteries. The placement of these catheters into veins and the bloodstream can have serious repercussions if they are done incorrectly (Hartley, 2024). These various catheter types are crucial in different medical scenarios and their accurate placement is vital for effective treatment and patient safety. The dataset provides a detailed analysis of these catheters placed in

different scenarios and patient profiles, allowing us to explore the factors that affect the accuracy of placement within these various patients.
This report aims to focus on factors that influence accurate placement of catheters and develop models to help doctors identify and understand the placement of catheters to a greater degree. In hospital and high-pressure situations doctors may make mistakes or miss key details. The purpose of this study is to find if the radiograph of a patient's catheter can be used by neural networks and models to support doctors and provide them with more information about the catheter's placement.

In order to do this, we can implement neural networks to analyse the radiographs and use this information to classify the catheters into their respective categories or extract key information about the catheters. The main type of neural networks this report will explore are convolutional neural networks (CNN). CNNs are currently the leading type of image recognition models used in the data science field for their efficiency and ability to segment and detect features within an image. CNNs are expected to perform extremely well, however the complexity of the radiographs and the computational cost of running models are expected to hinder the model's performance. Also, catheters being classified as borderline can only be a few cm of difference between a correct placement. Therefore, the models may not be able to model to detect these slight differences.

Exploratory Data Analysis

After conducting any preprocessing checks, a thorough data analysis was conducted to identify any important relationships between the different variables in the dataset. This led to making more informed decisions about what models can potentially be used for model development and what important variables should be the key focus.

What are the overall distributions of catheter placement statuses across different catheter types?

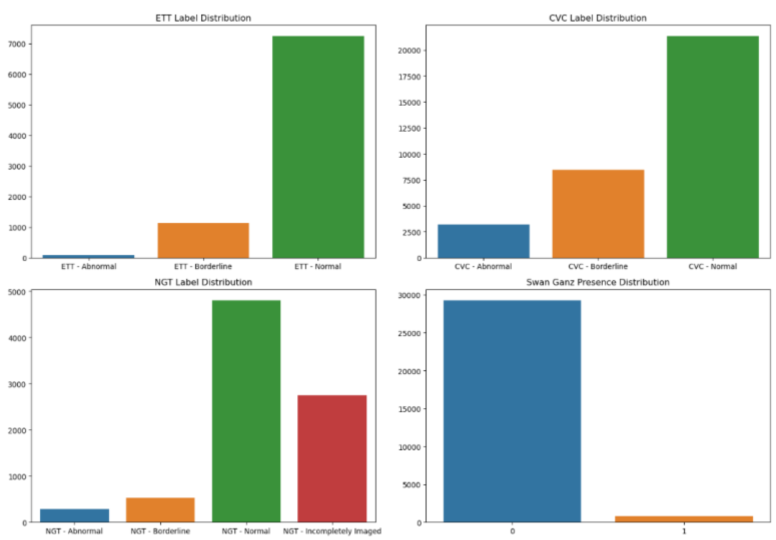


Figure 3: The distribution of different catheter types across the dataset

In figure 3, it was evident that the vast majority of endotracheal tube (ETT) placements are normal, with borderline and abnormal placements being relatively uncommon. Additionally, Central venous catheter (CVC) placements are mostly normal, but there is a notable number of borderline and abnormal placements. The distribution shows a more balanced spread between normal and abnormal cases compared to ETT. For nasogastric tube (NGT) placements, a large portion of the data is either normal or incompletely imaged, with abnormal and borderline cases being quite rare. Moreover, it was highly evident that most patients do not have a Swan Ganz catheter, with only a small proportion indicating its presence.

Overall, CVC represented the majority with 32,979 recorded placements significantly higher than the other catheter types. This suggests that CVCs are highly used in medical care due to their versatile use in accommodating various medical conditions. However, given their complexity and the significant high rate of incorrect placements observed, it should be a priority to improve the precision of these catheters.

ETTs and NGTs have more moderate frequencies, with 8,457 and 8,353 placements, respectively. This highlights their common use in managing patient airways and digestive systems. Their relatively lower count compared to CVCs may indicate that their use is less common and is only needed to provide respiratory support or feeding interventions.

Lastly, the Swan Ganz catheter only had 830 placements – highlighting it main use for only monitoring heart and lung function. This catheter didn't provide much insight into our further model development, hence CVV, ETT and NGTs were noted as the primarily catheters for future models.

How does the frequency of normal, borderline and abnormal placements differ among the different types of catheters [ET, NGT, CVC]

	Partial/Incorrect Count	Total Count	Percentage of Partial/Incorrect Placement
ETT	1217	8457	14.390446
NGT	808	8353	9.673171
CVC	11655	32979	35.340671

Figure 4: The frequency of partial/incorrect placements of different catheters amongst patients.

The table in Figure 4 highlights the performace of different across the different catheters: ETT, NGT, and CVC. ETT placements have a relatively low rate of incorrect placements at 14.39%, suggesting that the procedure for placing endotracheal tubes is generally effective, though some attention is still needed to reduce this error rate further. NGT placements displayed the lowest error rate at 9.67%, which is extremely favourable, but since nasogastric tubes are commonly used for feeding or drainage, even these minor placement errors can lead to complications such as feeding into the wrong location – this can cause life-threatening complications. CVC placements present the most concerning findings, with a 35.34% error rate, meaning more than one in three placements is either partial or incorrect. Central venous catheters are used for critical tasks like administering medication or monitoring hemodynamics, so any errors in placement could pose serious risks to patient safety. By analysing the high error rate among the different catheters, there is need for better training, procedural enhancements, and more advanced imaging techniques to ensure correct placement. Overall, CVC placements had the most incorrect/partial placements which indicates that the main focus should be on improving the accuracy of CVC placement to improve patient safety and outcomes. Although the error rates for ETT and NGT placements are low compared to CVC, it still should be encouraged to minimise incorrect placements as it should not be neglected to establish patient safety.

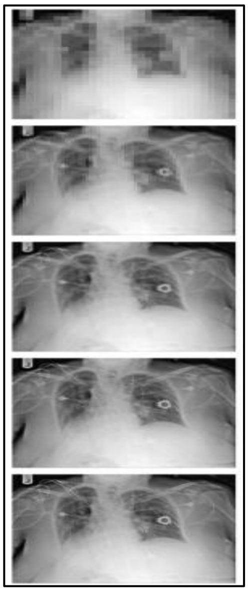


Figure 5: The greyscaled images over different pixel depths.

Next, we decide to check on how the images would look scaled to various size, as they will later have be passed through into the neural network at a set size. To compare the scaling of the images and their impact of the reshaping of the image to 32,64,128, 256, 512, 1024, 2048 along each axis of the image can be seen in figure 5. The results of this show that the greater number of pixels in the image the more detail can clear the image. From the results we would not use under 256 pixels for the training of the models, as the images lack clarity, and catheters cannot be accurately viewed. However, using the biggest image size may not always be the best. This is because the array size of the image is exponentially larger, leading to longer training and testing times. Another impact is that with larger image sizes more noise and unnecessary detail is captured. This can lead to models failing to recognise the catheter, or learning the incorrect features.

Model development and Results

Neural Network models can have a significant impact on the medical field, more specifically they can be a tool for doctors to use to improve their practice. Our goal is to create effective models that can potentially be applied to hospitals and clinics to help staff identify incorrect catheter placements, and where the catheters are placed. These models are not expected to replace the role of a doctor, but act as a means of support, aid, and reassurance that their job has been completed correctly and that the patient is in a non-critical and safe condition.

Analysis of correct and incorrect catheter placement using YOLOv8

The question that we want to answer firstly is whether we can determine if a catheter is correctly inserted or not. The result of doing this is that we can help doctors and hospitals to ensure and double check whether they have placed their catheters in patients correctly.

To answer this question, we need to initially format the images in the correct format and then move the images into the correct folders for training and testing of different models. The formatting of the images can involve many things. For the models in these circumstances two alterations were made to all the images and their data. These were there scaling down of the values in the image array by 255 and the resizing of all the images to fit the required image size of the models. The three image sizes that we investigated, and their performance are 256, 512, and 1024 pixels. There is one other optional formatting that needs to be done depending on the model. Some models can only take in images with rgb data arrays and some only grayscale. If the model required grayscale, then that modification was also applied when passing through the image into the model. There was one major issue when running models and that was the resizing of the images. Resizing the images before passing the image into the model was very time consuming, and thus exponentially increased the training time, as every image passed in had to be changed to a specific size. To fix this problem we used Microsoft's resize image application to resize all the images to the appropriate size before running the model (Microsoft, 2024). This meant that each image only needed to be resized once and not every time the image was passed into the model during the training process.

The next challenge was to label all the images for training the model. This process involved taking the image name, and the image name in the train cv and connecting the two. From here the images can then be moved into a ‘normal’ and ‘not_normal’ folder depending on whether their catheter is placed correctly or whether it has any errors of any type respectively. Once this was done the next step was to then randomly move a portion of these images into a train, validate, and test folder, each containing the label folders inside. The train folder had 70% of the images, validated 10% and 20%. Now that the folders were set up and the image data in the correct data, we can now progress to training models for the binary classification of whether a catheter is correctly placed or not. When testing the models, we want to explore which commonly used image classification models performed the best for this data set, but also how factors such as image size can affect the performance of the models.

After researching the current use of neural networks in medical image classification, we found that two models were frequently used and performed well, Yolov8 and ResNet50 performed well on cancer classification (Patel, Kanojia, & Nair, 2024). The research into the use of image classification in the medical field showed that it is a practical and an accurate method to detect and improve the detection of diseases and help improve hospitals performances. The first model that we chose to explore was yolov8. Yolov8 is primarily an object detection model, however they have released image classification models to be used. Yolov8 classify offers a range of models: nano, small, medium, large, and extra-large. However, for this we will only focus on the small, medium and large models for the classification. Yolov8 is known for their fast and real time implementation of object detection and classification. It does this through using a

Convolutional Neural Network, specifically CSPDarknet53. This involves using cross stage partial networks, involving the feature map being split into two sections and one section going through several convolutional layers and the other merging after the layers (Torres,2024) . Doing this addresses two problems.

It firstly addresses the vanishing gradient problem, the problem that when backpropagation is used on neural networks the earlier layers' the loss function's gradients become so small it is difficult to adjust and therefore slows down the loss functions convergence. To fix this CSP uses the connection of earlier layers to later layers to ensure that the gradients are still being Passed through the network and not becoming insignificant (Engati).

The other problem that CSP addresses is memory and speed of the network. By splitting up the feature map into two, it saves memory and computation time. All this makes Yolo classification a great object and classification model, with great feature extraction. Yolo's biggest positive compared to all the other models tested is its speed of real time classification, making it a great choice for real world application of placement catheter classification.

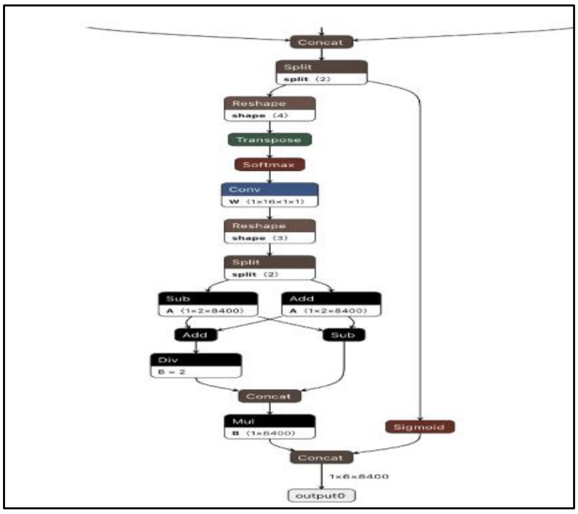


Figure 6: A visual representation of a cross stage partial network.

There are 5 possible yolov8 classification models to choose from. Each model comes with a different number of parameters and deployment speed. Each model has also been tested on 'ImageNet', a popular image classification dataset. From this dataset, it is clear that the more parameters the better accuracy they achieved. However, due to computational complexity of running the xl model, we decided to not include it in our testing.

Different Yolov8 models available

Model	size (pixels)	acc top1	acc top5	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)
YOLOv8n-cls	224	69.0	88.3	12.9	0.31	2.7
YOLOv8s-cls	224	73.8	91.7	23.4	0.35	6.4
YOLOv8m-cls	224	76.8	93.5	85.4	0.62	17.0
YOLOv8l-cls	224	76.8	93.5	163.0	0.87	37.5
YOLOv8x-cls	224	79.0	94.6	232.0	1.01	57.4

Figure 7: A tool to help identify the different YOLOv8 models.

We want a model that allows doctors and nurses the ability to double check their patients and support them. Also because of the seriousness of the classification, being someone's health, we want to choose the model with the lowest false negative rate. This means we want a model that has the highest chance of predicting if a catheter is not in place properly. We can then test all the models for the number of false negatives they get, to initially see which model may be the best. As seen in the figure, the models that take in a higher image resolution perform much better than a lower one. This is expected, as seen previously in the exploratory data analysis the higher the image's resolution is the more detail is in the image and the easier it is to distinguish the catheter from bones and other body parts in the images. Comparing Yolov8's models between each other, Medium proved to have the lowest false negative, with 67.6% of all the not normal catheters being detected. Then the large model performed the second best, then the small model. The large models despite having the most layers and parameters did not perform the best. This is likely due to the overparameterization. This means that having too many parameters is leading to overfitting, due to noise of the training dataset being captured. However, 67.6% of incorrectly placed catheters being identified is low, considering the health implications that can occur if a catheter is not placed correctly.

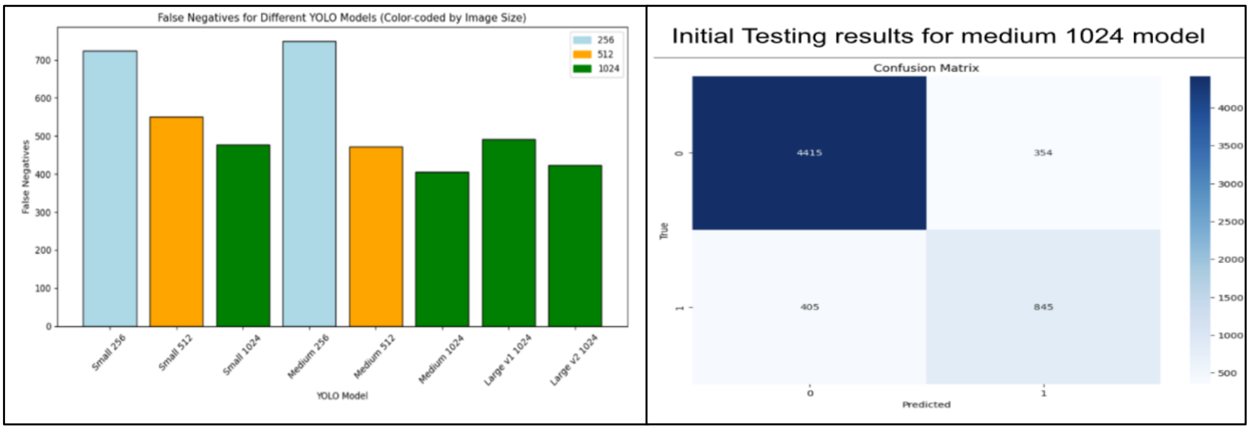
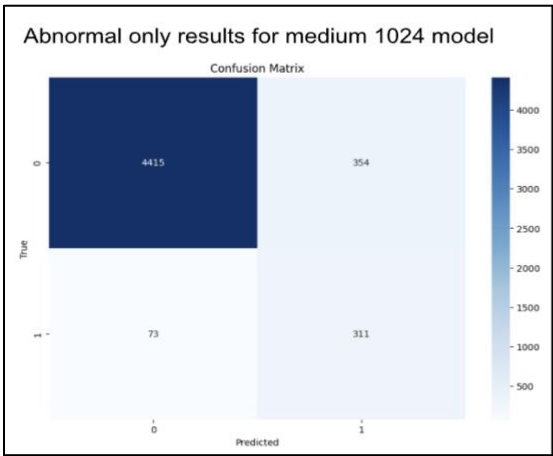


Figure 8: Left shows the false negatives for YOLO models depending on pixel depth. Right shows the initial testing results confusion matrix for medium YOLO model with 1024 x 1024 pixels.

To see where the model is having trouble predicting the catheters, we can limit the catheters that are being tested to only catheters that are completely abnormal and not the catheters that are partially. Doing so drastically improves the model's performance in terms of false negative rates, with 80% of abnormal catheters now being detected for the 1024 medium model. This shows that the model is able to distinguish between a correct and incorrect catheter majority of the time, but struggles with the small details that make a catheter partially incorrect.



Measure	Value
Sensitivity	0.9837
Specificity	0.4677
Precision	0.9258
Negative Predictive Value	0.8099
False Positive Rate	0.5323
False Discovery Rate	0.0742
False Negative Rate	0.0163
Accuracy	0.9171
F1 Score	0.9539

Figure 9: Left shows the abnormal only results for medium YOLO model over 1024 x 1024 pixels. Right shows the relative accuracy scores.

Secondly, we can analyse why the large model did not outperform the medium model, despite having 15 million more parameters. The reason why this is happening can be seen with the training and validation information for the large model. Here the train loss is continuing to decrease for each epoch, however the validation loss plateaus, indicating that there is overfitting of the data, or that there are too many parameters in the model for this task. Comparing this to the research paper that used Yolov8 for the classification of breast cancer, the results are very different. Though breast cancer and catheters both belong to the medical field, there is a stark difference between the two. As seen in the image of the breast cancer training image sample, this looks completely different compared to the catheter images, and there we should not expect the same results when reproduced. Secondly, one of the reasons that Yolov8 might not have achieved the same results as seen in this study is that the catheter being incorrect is detected through where the catheter is seen in the image and where it is going. This is different to the breast cancer problem, as Yolov8 tries to create a bounding box around the cancerous cell. For the catheter trying to create a bounding box around them would be very difficult as many curves and travel across the body. At the current stage this Yolo model for incorrect catheter detection can only detect completely abnormal catheters, and because of this it would need further refining and training if it can be used for real medical practice.

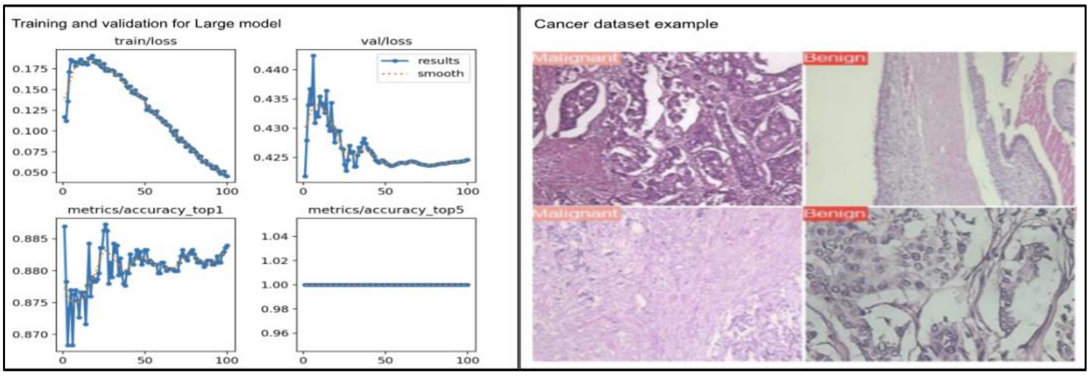


Figure 10: Left shows the training and Val visualisations for large model. Right shows cancer dataset example.

Annotating of catheters placement using UNet

Another use of modelling is to help highlight where the catheters are placed for the doctors. This will allow doctors to have a clearer indication of where the catheter is placed and therefore allow doctors to have an easier time analysing whether the catheters are placed correctly. In order to create an annotation of the catheter, we used the CNN, UNet.

UNet is a very important Convolutional Neural Network (CNN) architecture that shows great potential especially in medical image segmentation and other tasks that require fine pixel-level prediction. Its importance lies mainly in its ability to segment images with high accuracy and its suitability for situations where the training samples are small, reflecting good generalisation capabilities.

Firstly, the architectural design of UNet makes it particularly suitable for image segmentation tasks. It adopts a symmetric ‘U’ structure, which consists of two parts: the encoder and the decoder. The encoder is used to extract high-level features from the image, while the decoder progressively reduces these features back to the original image size and performs pixel-level classification. This symmetric network structure enables UNet to capture the relationship between the global information and the details of an image, ensuring that both the overall contextual information and the fine features of the details are preserved in the segmentation process.

Secondly, the skip connections in UNet are one of the key factors for its success. These skip connections are able to combine the features in the encoding stage with the corresponding parts in the decoding stage, thus solving the problem that information may be lost in the down sampling process. This not only improves the segmentation accuracy, but also allows the model to perform better in terms of boundaries and details, especially in the case of complex backgrounds or irregular shapes.

Firstly, to be able to use UNet, the images need to firstly be pre-processed. For the training images, we use all the images in the dataset that already had annotations stored as coordinates. During image processing, we use some preprocessing and augmentation techniques to improve the performance of the model. In this project, we perform resize operation and image enhancement. Image enhancement consists of four parts: rotate horizontally, adjust brightness, adjust contrast, and add noise. When applying these image enhancement operations, we treat them as a loop, where each operation has a 50% probability of being triggered. This increases the diversity of the training data and improves the robustness of the model.

After the augmentations we have 30,000 images, and we have resized them to 512x512, with the training set containing 17,999 of them. An initial problem that we noticed is that the coordinates of the catheter's lines need to also change during resize and horizontal rotation operations, and the model needs to be able to adapt to these transformations to better understand important features in the images. After implementing this change, the model (UNet) is able to learn the features more efficiently under different shapes and conditions, thus improving the segmentation accuracy.

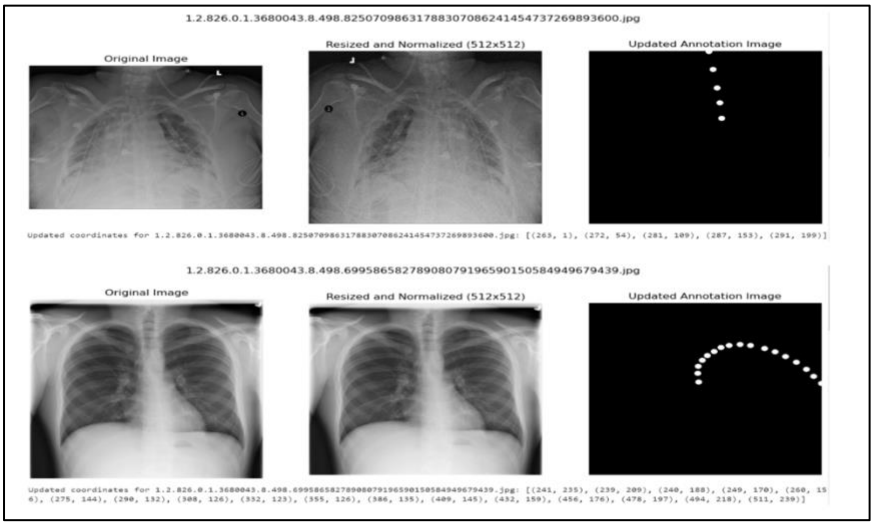


Figure 11: Comparisons between original images, augmented images and updated annotation images from preprocessing

In addition, we use the python library CV2 to put the pipeline of images on display. In an image of size 512x512, it is possible that the catheter pipeline is not very obvious, so we expand the radius of the points of the pipeline and connect these points to form a basic lane structure (basic lane). This gives a better presentation of the pipe features in the image and thus helps the model to learn the relevant feature information more efficiently. Doing so will also make it easier for doctors to visualise and analyse a catheter's placement.
The dataset now consists of pre-processed images and corresponding annotations. Images were stored in a NumPy array for efficient loading, and annotations were provided in a CSV file used to create binary masks representing the regions of interest. The images and masks were then normalised to values between 0 and 1 to standardise input data for the model. The dataset was split into training and validation sets with an 80-20 ratio to ensure a balanced evaluation of the model.

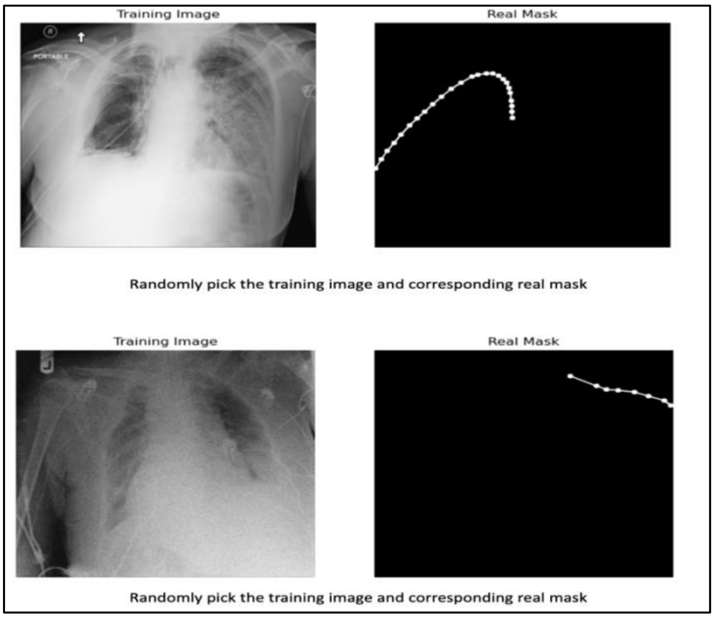


Figure 12: Randomly pick the training image and the corresponding real mask.

The model we used is based on a U-Net architecture, commonly used for image segmentation tasks due to its symmetrical encoder-decoder structure. The encoder extracts feature through successive convolutional layers and max-pooling operations, gradually reducing spatial dimensions while increasing feature depth. The decoder reconstructs the segmentation mask by up sampling the feature maps and concatenating them with corresponding encoder outputs, which helps retain spatial information. The proposed model includes an attention mechanism in the bottleneck layer. The attention module uses global average pooling and global max pooling, followed by a dense layer with a sigmoid activation, to create a spatial attention map. This attention map is used to weight the feature maps, emphasising important regions and suppressing irrelevant features, thereby enhancing segmentation accuracy.

The model uses a combination of binary cross-entropy and Dice loss to improve segmentation accuracy. Binary cross-entropy is effective for pixel-wise classification, especially when dealing with imbalanced datasets. Dice loss improves sensitivity to small regions, which is crucial for accurately segmenting the annotated catheter points.

The model was trained using the Adam optimizer with a learning rate of 0.001. The Adam optimizer was chosen for its ability to adapt the learning rate during training, providing efficient convergence. An early stopping mechanism was used to avoid overfitting by monitoring the validation loss, with training halted if no improvement was seen for 5 consecutive epochs. The batch size was set to 4 due to memory constraints, and the model was trained for a maximum of 15 epochs.

To verify preprocessing, a random selection of training images and their corresponding masks was visualised. The images and masks showed good alignment, highlighting the catheter and other key features, indicating that the preprocessing steps were correctly applied.

Using the combined binary cross-entropy and Dice loss improved the model's sensitivity to small regions of interest. The attention mechanism enabled the model to better focus on important features during the bottleneck stage, improving segmentation compared to a standard U-Net. The training and validation loss curves demonstrated that the model effectively learned meaningful features from the data, with early stopping helping to select the best-performing model based on validation loss, thereby preventing overfitting.

In summary, we generated a U-Net model with an attention mechanism developed for medical image segmentation, specifically for catheter localization. The attention module improved the model's ability to focus on relevant features, enhancing segmentation performance. The combined loss function of binary cross-entropy and Dice loss was effective in improving the detection of small, annotated areas. The model was trained on a CPU, and the results showed a validation accuracy of approximately 98% and a validation loss of around 0.46. These results are very good given the computational limits of training on a CPU compared to a GPU. This model can be implemented into hospitals and be used by doctors to help find and identify catheters more clearly. Another possibility for further improvement is using the outputs of this model to help train the two previous models. The annotations of the catheter are likely to increase the accuracy and performance of the models, which is something that we would like to test in the future.

Technological improvements of the UNet model

One of the main limiting factors that diminishes the findings of the models is the computational cost to train the models. Most of the models used in this report used image sizes above 512x512, and used 3 colour channels (RGB) for the images. This comes to 786,432 values per image. Considering that for training the model well over 15,000 images were used, and that is a lot of processing time. Secondly, CNN's such as UNet, ResNet, DenseNet, and Yolov8 have a large number of trainable parameters, with some models exceeding 30 million. Due to this, the training of these models on a CPU is just not feasible, or if it may not give optimal results, due to the time needed to complete the relevant computations and low batch size and epoch count used. Moreover, even using a low-level GPU or TPU did not provide sufficient training for these models as the VRAM needed for batch sizes over 8 was too much for even those (VRAM needed was above 16GB). Therefore, for optimal training to be done on these large models a better GPU may result in better performance. The issue arose for us in that the GPU's such as Nvidia's A40, which would have allowed the running of larger models, were erroring due to outdated CUDA software and not being compatible with the current TensorFlow version. Without access to the terminal the GPU is based on, CUDA cannot be updated and therefore libraries such as TensorFlow and pytorch cannot train on the GPU. This meant all the training for our modelling occurred on the CPU and likely resulted in suboptimal performance/results.

Utilization of ResNet and DenseNet

We further utilised Resnet and DenseNet to explore the accuracy of models in classifying labels correctly to image. The first model was tested using the resnet50 function contained in the keras.applications library. ResNet (Residual Networks) is an ideal choice here because it excels at image classification tasks and handles vanishing gradients better due to its residual connections. This makes ResNet highly reliable for classifying complex medical images with precision.

Furthermore, the AUC score was used instead of accuracy for this as it serves as a better fit for data with class imbalance and as mentioned in the EDA section ,the CVC normal was significantly higher than the rest and the other reason for using AUC score was that in medical scenarios, false positives and false negatives can have very different consequences. For instance, a false negative (predicting a wrong placement as correct) can be far more dangerous than a false positive (predicting a correct placement as wrong). AUC allows you to focus on the model's true positive rate (sensitivity) and false positive rate (specificity), which are crucial in something like healthcare. In medical applications, it's essential to ensure the model is not just "right" most of the time (which accuracy measures), but that it can **confidently distinguish** between correct and incorrect outcomes (which AUC measures).

```
history=model.fit(train_datagen, epochs=8, batch_size=BATCH_SIZE,
          steps_per_epoch=len(train_datagen),
          validation_data=valid_datagen,
          validation_steps=len(valid_datagen),
         )
Epoch 1/8
754/754 [==============================] - 3233s 4s/step - loss: 0.3641 - binary_accuracy: 0.8537 - auc: 0.8049 - val_loss: 0.2
944 - val_binary_accuracy: 0.8846 - val_auc: 0.8618
Epoch 2/8
754/754 [==============================] - 3155s 4s/step - loss: 0.2999 - binary_accuracy: 0.8851 - auc: 0.8576 - val_loss: 0.2
959 - val_binary_accuracy: 0.8846 - val_auc: 0.8617
Epoch 3/8
754/754 [==============================] - 3171s 4s/step - loss: 0.3007 - binary_accuracy: 0.8851 - auc: 0.8572 - val_loss: 0.2
960 - val_binary_accuracy: 0.8846 - val_auc: 0.8703
Epoch 4/8
754/754 [==============================] - 3205s 4s/step - loss: 0.3006 - binary_accuracy: 0.8851 - auc: 0.8581 - val_loss: 0.2
952 - val_binary_accuracy: 0.8846 - val_auc: 0.8661
Epoch 5/8
754/754 [==============================] - 3205s 4s/step - loss: 0.3015 - binary_accuracy: 0.8851 - auc: 0.8576 - val_loss: 0.2
955 - val_binary_accuracy: 0.8846 - val_auc: 0.8643
Epoch 6/8
754/754 [==============================] - 3183s 4s/step - loss: 0.3023 - binary_accuracy: 0.8850 - auc: 0.8571 - val_loss: 0.2
954 - val_binary_accuracy: 0.8846 - val_auc: 0.8691
Epoch 7/8
754/754 [==============================] - 3039s 4s/step - loss: 0.3018 - binary_accuracy: 0.8848 - auc: 0.8589 - val_loss: 0.3
014 - val_binary_accuracy: 0.8846 - val_auc: 0.8593
Epoch 8/8
754/754 [==============================] - 3050s 4s/step - loss: 0.3015 - binary_accuracy: 0.8845 - auc: 0.8599 - val_loss: 0.2
961 - val_binary_accuracy: 0.8846 - val_auc: 0.8712
```

Figure 13: ResNet model over 8 epochs

8 epochs were ran to train the model, totalling around 8 hours to fully train. After training the model, the summary function was used to find the AUC Score for the resnet50 function. Using this, the graph was created to illustrate the results found:

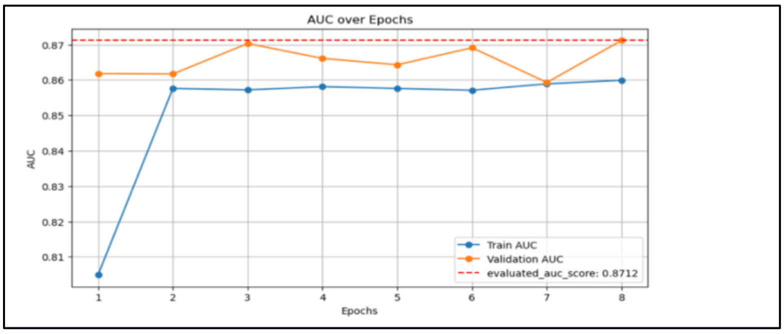


Figure 14: A strong visualisation of AUC over Epochs

What is the important to note here is the evaluated AUC score which was the number obtained from evaluating the model and the how the model just plateaus really after the second epoch. When the model plateaus like this, the reasoning is most likely that the model has converged, meaning it has reached its optimal performance given the current data, architecture, and training setup. However, the AUC score produced by the Resnet50 function produced a high result at around 87%, which was in line with what we expected as Resnet is universally accepted as one of the best image classifiers out there. The results reflect the conclusion that Resnet is reliable in classifying the catheters into its correct labels.

Furthermore, a decision was made to train another model to compare and contrast with Resnet to obtain even further confirmation that Resnet was the best possible model for this task. To achieve this, Dennsenet121 also from the karas.applications was utilised and trained to obtain results and compare with Resnet. Similar code and training were done to reduce any external factors that may impact the results:

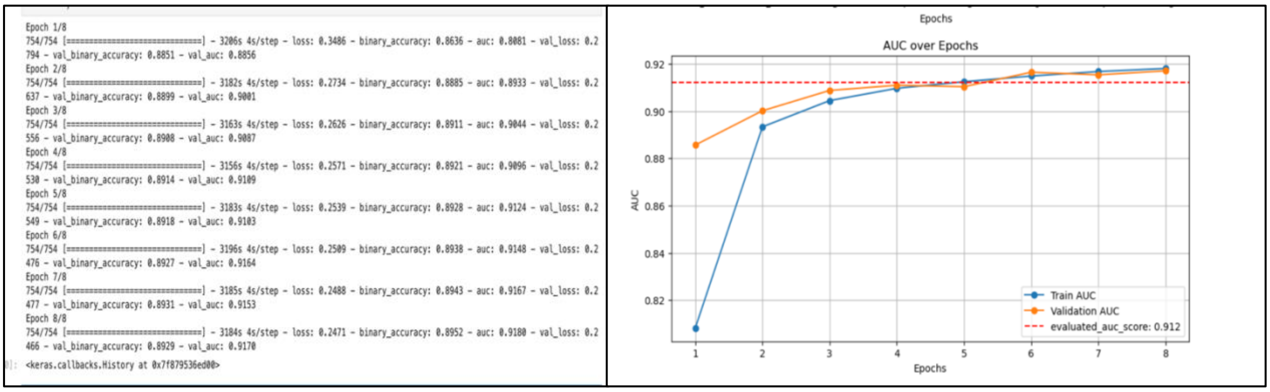


Figure 14: Left is the DenseNet scores. Right, AUC over epochs visualisation using DenseNet.

Contrary to expectation, the Densenet121 model actually returned a higher AUC score compared to the Resnet50 model at around 91%. There were two main reasons we suspected that may have cause these unexpected results. The 2 reasons were Parameter Efficiency and gradient flow**:**

- DenseNet tends to be more parameter-efficient than Resnet. Although DenseNet has more connections, it requires fewer parameters because layers can reuse previously learned features instead of learning redundant ones. Resnet, while powerful, may require deeper networks or more parameters to achieve the same level of performance, which could lead to overfitting or less efficient training
- DenseNet in general also tends to have stronger gradient flow, which helps it learn efficiently in fewer epochs and capture important features that Resnet may have missed.

Also important to note is that unlike Resnet, the results did not plateau as we added on more epochs which suggested the accuracy could go higher, but due to technical problems, 8 epochs was the max number of epochs that could have been trained, therefore leaving the true max score for the DenseNet function unobtained.

Conclusion

In order to enhance the analysis and classification of catheter placements in medical pictures, we investigated the application of many deep learning models in this research, including UNet, ResNet, and DenseNet. Every model fulfilled a distinct function, tackling various obstacles associated with the segmentation and classification of medical images.

In order to precisely segment catheter sites in radiographs and enable physicians to clearly see the placements, the UNet model was essential. The UNet enhanced the model's capacity to identify minute features and manage complicated backdrops by utilising a symmetric encoder-decoder architecture with skip connections and augmentations like brightness modification and rotation. By highlighting essential characteristics, the attention mechanism improved segmentation performance even further. The model demonstrated its efficacy even with limited resources, as seen by its 98% validation accuracy, even with training constraints on a CPU.

AUC was used as the assessment metric to address class imbalance and evaluate the sensitivity and specificity of the ResNet50 model—two crucial components in healthcare applications—while classifying catheter placements. ResNet performed well, with an AUC of 87%; nevertheless, the model's performance plateaued early, suggesting that, with the available data and configuration, it had reached convergence.

The same dataset was used to train DenseNet121 in order to compare and validate these outcomes. It surprised everyone by outperforming ResNet with a 91% AUC score. Dense Net’s performance may be ascribed to enhanced gradient flow and parameter efficiency, which let it capture important characteristics more successfully in fewer epochs. Additionally, DenseNet’s performance did not plateau, suggesting the potential for even better results with extended training, though hardware limitations prevented further exploration.

In conclusion, each model—UNet, ResNet, and DenseNet—played a complementary role in the project. The UNet model excelled at segmentation, offering precise annotations of catheter placements that can assist doctors in visual analysis. ResNet and DenseNet proved valuable for classification, with DenseNet showing greater potential for long-term improvements. This project highlights how deep learning models can address both segmentation and classification challenges in medical imaging. Moving forward, implementing these models on more powerful hardware, such as GPUs with updated CUDA support, could further enhance performance. Additionally, integrating the UNet’s outputs as features for ResNet and DenseNet may boost their classification accuracy, creating a synergistic pipeline for catheter analysis.

References:

Engati. (n.d.). *Vanishing gradient problem*. Engati. https://www.engati.com/glossary/vanishing-gradient-problem

Microsoft. (2024, July 12). *PowerToys Image Resizer utility for Windows*. Microsoft.com. https://learn.microsoft.com/en-us/windows/powertoys/image-resizer

Torres, J. (2024, January 15). *YOLOv8 Architecture: A Deep Dive into its Architecture - YOLOv8*. Yolov8. https://yolov8.org/yolov8-architecture/

Vaibhav Patel, Mahendra Kanojia, & Vainavi Nair. (2024). Exploring the Potential of ResNet50 and YOLOv8 in Improving Breast Cancer Diagnosis: A Deep Learning Perspective . *International Journal of Computer Information Systems and Industrial Management Applications*, *16*(3), 16. Retrieved from https://cspub-ijcisim.org/index.php/ijcisim/article/view/717

Brown, S. (2018, February 21). *What Are the Types of Catheters?* WebMD; WebMD. https://www.webmd.com/urinary-incontinence-oab/catheter-types

Hartley, L. (2024, October 14). *Lines and tubes (radiograph) | Radiology Reference Article | Radiopaedia.org*. Radiopaedia. https://radiopaedia.org/articles/lines-and-tubes-radiograph

Royal Australian & NZ College of Radiologists. (2024). *RANZCR CLiP - Catheter and Line Position Challenge*. @Kaggle. https://www.kaggle.com/c/ranzcr-clip-catheter-line-classification

Engati. (n.d.). *Vanishing gradient problem*. Engati. https://www.engati.com/glossary/vanishing-gradient-problem

Microsoft. (2024, July 12). *PowerToys Image Resizer utility for Windows*. Microsoft.com. https://learn.microsoft.com/en-us/windows/powertoys/image-resizer

Torres, J. (2024, January 15). *YOLOv8 Architecture: A Deep Dive into its Architecture - YOLOv8*. Yolov8. https://yolov8.org/yolov8-architecture/

Vaibhav Patel, Mahendra Kanojia, & Vainavi Nair. (2024). Exploring the Potential of ResNet50 and YOLOv8 in Improving Breast Cancer Diagnosis: A Deep Learning Perspective . *International Journal of Computer Information Systems and Industrial Management Applications*, *16*(3), 16. Retrieved from https://cspub-ijcisim.org/index.php/ijcisim/article/view/717