Image Guidance for Robot-Assisted Ankle Fracture Repair

Critical Review paper: NAS-Unet: Neural Architecture Search for Medical Image Segmentation

by Jayaram Mandavilli

Group 10

Team Members: Asef Islam and Tony Wu

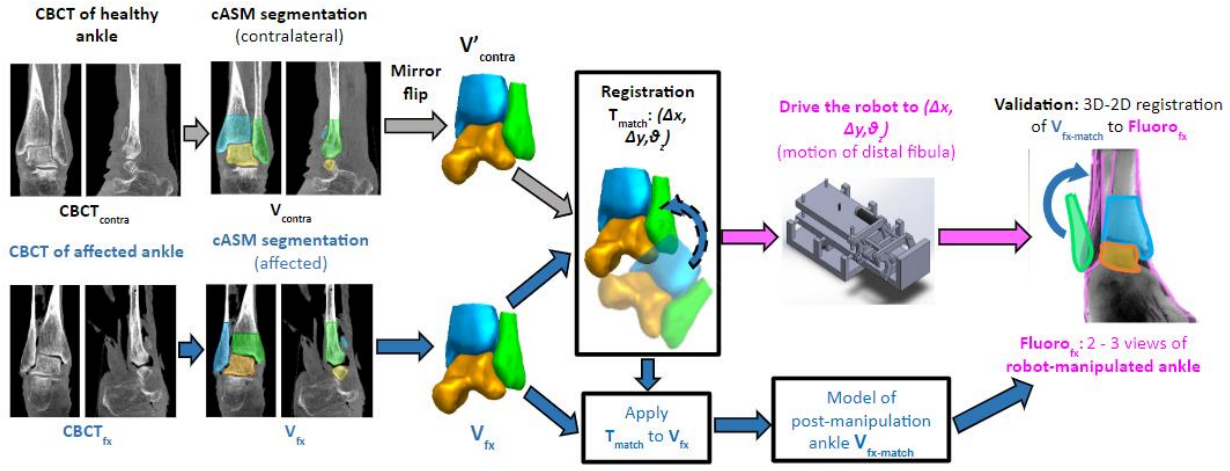
Mentors: Dr. Jeffrey H. Siewerdsen and Dr. Wojtek Zbijewski

Project Summary:

Our project involves surgical repair of ankle during fracture. Many fractures cause dislocation of the fibula. This dislocation disrupts the syndesmosis because it forcefully displaces ligaments and connective tissue in the syndesmosis. In order to restore the syndesmosis, the fibula and fixation need to be placed at the correct position. However, the current standard of care is the surgeon has to manually do this just by estimating the locations. The issue is when this placement is not done perfectly is it can lead to PTOA (Post Traumatic Osteoarthritis). It has also been found that 70% of ankle fracture cases lead to PTOA mainly because of this reason. This project is exploring an image guided approach to minimize this issue.

Technical Approach:

Multiple approaches are being looked into in order to properly analyze images. Work has already been done to show that high quality Cone Beam CT (CBCT) images can be segmented and used in this project. However, this has not been done for lower quality C-arm images. By applying this problem to C-arm images, it will allow the process to become intra-operative. This way an image of the patient's ankle can be taken and then immediately the injured ankle image can be segmented and then registered to a healthy ankle. Right after that, instructions can be given to the robot in order to place the fibula in a much better position than if the surgeon just had to estimate. The diagram below shows the overall workflow of this project.



This project is focusing on the initial steps of this project where the images are automatically segmented. In order to do this, two approaches are being attempted. The first is the use of a coupled Active Shape Model (cASM). The other approach involves deep learning. Multiple approaches are being looked at so that ultimately the one with the highest segmentation accuracy can be used. It is also possible to use a combination of these models were a neural network would be used to learn where to initialize the cASM and the cASM would work from there.

Paper Choice:

The deep learning approach is very interesting because there are a lot of possibilities. Either it can be used to do the entire segmentation or used in tandem with the cASM to increase the accuracy of

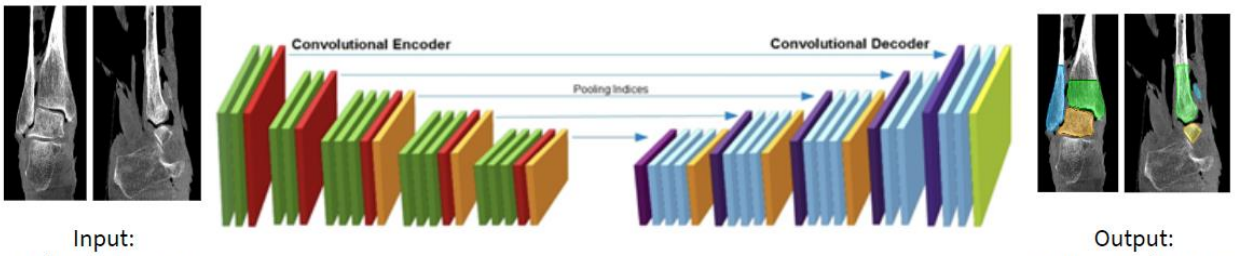
the sole cASM model. The paper below shows a neural network architecture used for medical image segmentation.

Weng, Y., Zhou, T., Li, Y., & Qiu, X. (2019). NAS-Unet: Neural Architecture Search for Medical Image Segmentation. *IEEE Access*, *7*, 44247–44257. doi: 10.1109/access.2019.2908991

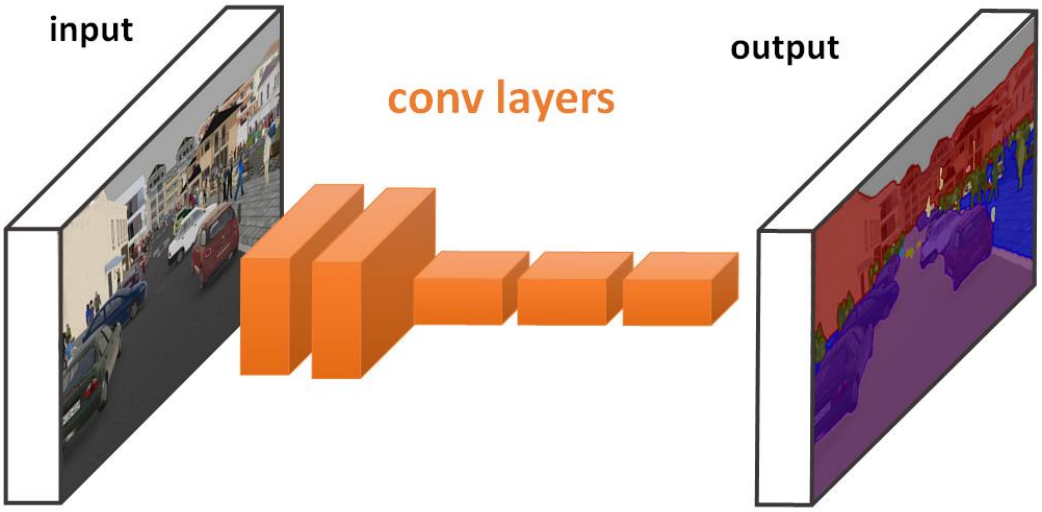
The reason this paper was chosen was because it explored a modification of the already established U-Net to medical image segmentation. This is an architecture that is designed to work with less training data.

Paper Background:

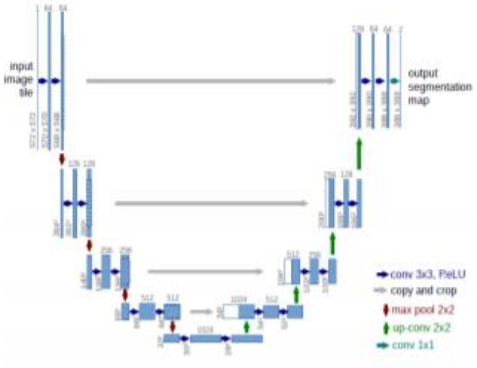
This paper builds on the already established U-Net architecture which is a type of Convolutional Neural Network (CNN). CNNs are commonly used for image segmentation both in the field of medicine and outside of it. This is because they tend to have the highest segmentation accuracies out of the common architecture types. CNNs are very useful for medical image segmentation because they can do semantic segmentation which involves pixel wise segmentation instead of larger scale segmentations within the image. At high level, they work by taking patches around each pixel so that they can classify that particular pixel. This allows them to produce a multi-channel likelihood map. CNNs also commonly use pooling (for example max pooling) strategies. This is a way they can alleviate the computational burden from training these networks. The basic structure of a CNN is to have downscaling layers and then upscaling layers. This way, the neural net reduces the number of pixels in the image by taking only those that are of interest. After that it is gradually able to add back pixels and label all of the pixels in the original image accordingly. The downscaling layers are called the encoder and the upscaling layers are called the decoder. The image below shows a typical CNN structure and the inputs and outputs if it were applied to our project.



Fully Convolutional Networks (FCNs) have also been looked into as alternatives to CNNs. CNNs have convolutional layers followed by fully connected layers. The issue here is that with the fully connected layers, the image dimensions have to be same for all training and testing images. This is why FCNs were looked at. In FCNs, the fully connected layers are replaced by one up sampling layers. Since this isn't a fully connected layer, FCNs can take in images with varying dimensions. The picture below depicts a typical FCN. The beginning of the network is similar to a CNN but with the difference of one up sampling layer in the decoder portion instead of multiple fully connected layers.



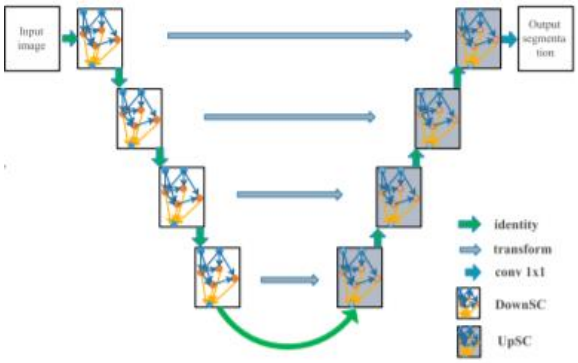
U-Nets are similar to FCNs in the way they are also a type of CNN. However, there are some significant differences. For example, U-Nets have multiple up sampling layers unlike the FCN which only has one. Another major difference is that U-Nets use skip connections to connect each pair of down sampling and up sampling layers. These connections are beneficial because it makes spatial information connect to deeper layers. This helps increase the segmentation accuracy. The picture below shows the U-Net structure. This is where this structure gets its name from.



Paper Purpose:

This paper explores a modification to the U-Net which they termed NAS-Unet. Their goal was to improve upon the U-Net in order to achieve a higher segmentation accuracy. They saw that the U-Net was very successful in medical image segmentation. However, they decided to use a machine learning approach to optimize the architecture in order to increase the accuracy. The biggest modification they made to the U-Net is they used a cell-based architecture. This meant that each block (as seen in the U-Net) was altered into a substructure by itself. They used convolution as well as other primitive operations to build these sub structures. Furthermore, they utilized machine learning to find the best substructures. They were able to look through possible structures and found one for the down scaling cells and one for the upscaling cells. Another modification to the U-Net architecture they made was they replaced the skip connections with cweight operations. They did this so that weights were assigned to feature channels. This way, they were able to show which features were redundant (lower weights and

less relevant) and which features were more significant (higher weights and more relevant). The basic structure of the NAS-Unet can be seen in the image below.



Paper Experiment:

This paper conducted three different experiments with their NAS-Unet. Each one involved different image formats and segmentation of different anatomical features. They were able to test both 3D and 2D images because they created the model so that it worked on 2D images. Thus, they took slices of the 3D images and fed these into the model as 2D images. They also tested a basic U-Net and a FC-densenet so that they could see if the NAS-Unet was able to perform better than these base line architectures.

When it came to selecting primitive operations for the cells in the NAS-Unet, they made sure that there were no layers in the cell that were the exact same. Thus, two layers in the same cell could have been the same operation but only if their parameters varied. They also designed the substructures to have less parameters. The table below shows all of the primitive operations looked at.

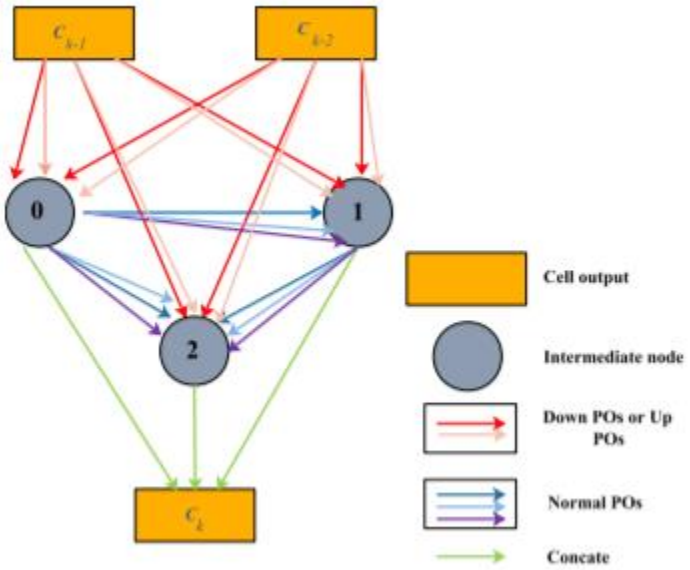
POs type	Down POs	Up POs	Normal POs
1	average pooling	up cweight	identity
2	max pooling	up depth conv	cweight
3	down cweight	up conv	dilation conv
4	down dilation conv	up dilation conv	depth conv
5	down depth conv	-	conv
6	down conv	-	-

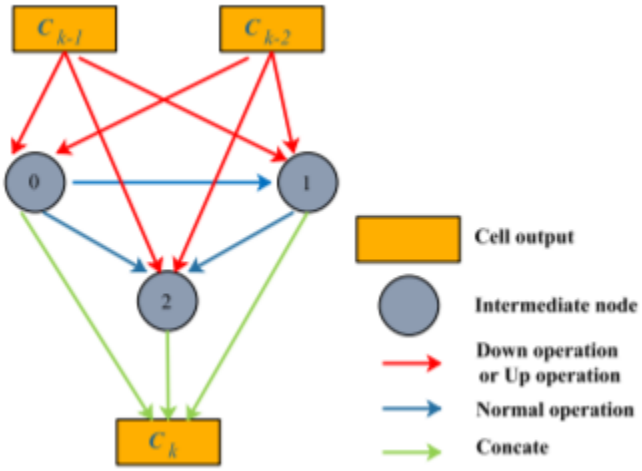
When it comes to their search strategy, they found a much more efficient operation parameter update strategy than an over parameterized one. This was important because it minimized the number of parameters in the NAS-Unet which reduced the computational burden.

Each edge in this substructure was called a mixed operation where the output would be described by the equation below. Their efficient substructures utilized soft maxes and minimized the number of overall parameters.

$$MixO(x) = \sum_{(i=1)}^{N} w_i o_i(x)$$

The first image below shows an overparameterized structure. The second one shows a regular structure.





Experiment 1:

The first experiment involved 50 patient prostates. The images for this part of the experiment were MRI images. MRIs as a whole are 3D images so in order to use if for the networks, they obtained 1250 slices from the images.

Experiment 2:

The second experiment involved two types of images. It looked at abdominal CT and MRI images. The goal of this part of the experiment was to segment livers in the CT images and other abdominal organs in the MRI images. Like MRI images, CT images are also 3D images. Thus, slices were once again taken. They had 40 patient CT images. From that they got 2874 slices for training and 1408 slices for testing. From the MRI images, they obtained 1594 slices for training and 1537 slices for testing.

Experiment 3:

The third experiment involved the segmentation of nerves in the brachial plexus. This part of the experiment utilized ultrasound images in the neural networks. These are 2D images so slices were not needed. They had 5635 images for training and 5508 images for testing.

Paper Results:

When it comes to analyzing the results of these experiments, an important metric called the mean intersection over union (mIoU). It is the mean of the IoU for each test image where IoU is represented by the equation below.

$$IoU = 100 * \frac{true-positives}{true-positives+false-negatives+false-positives}$$

The tables below the results for each experiment and each of the architectures tested. The mIoU is the most significant statistic because it shows how accurate each network was

Experiment 1:

Model Type	mIoU	DSC	Train Time	GM
U-Net	0.978	0.938	**6h**	**1.3**
FC-Densenet	0.982	0.956	23h	2.7
NasUnet	**0.983**	**0.9737**	8h	1.5

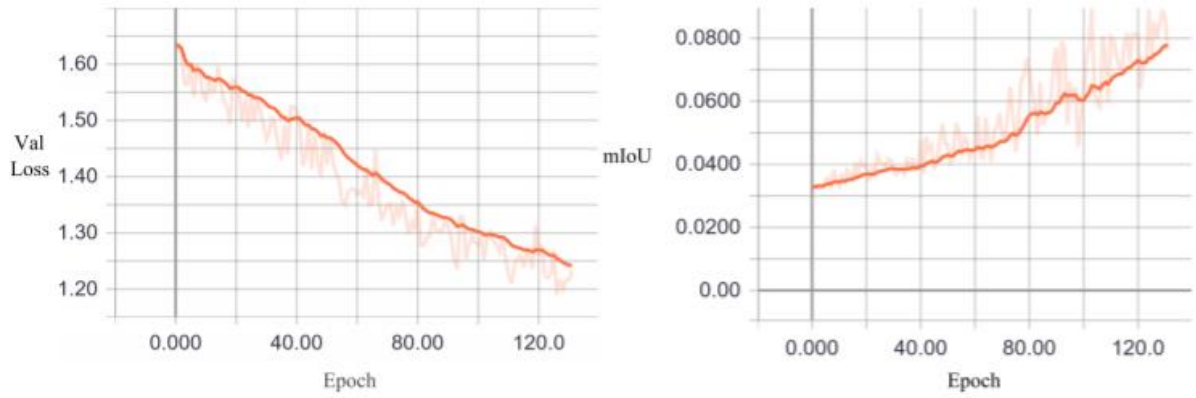
Experiment 2:

Model Type	mIoU	DSC	Train Time	GM
U-Net (CT)	0.982	0.937	**1d-6h**	**2.4**
FC-Densenet (CT)	0.983	0.965	3d-4h	6.84
NasUnet (CT)	**0.985**	**0.974**	1d-15h	3.5
U-Net (MR)	0.46	0.682	**9h**	**1.3**
FC-Densenet (MR)	0.51	0.734	21h	2.7
NasUnet (MR)	**0.54**	**0.76**	11h	1.5

Experiment 3:

Model Type	mIoU	DSC	Train Time	GM
U-Net	0.989	0.74	**18h**	**2.3**
FC-Densenet	0.989	0.844	2d-3h	6.85
NasUnet	**0.992**	**0.881**	1d-3h	3.4

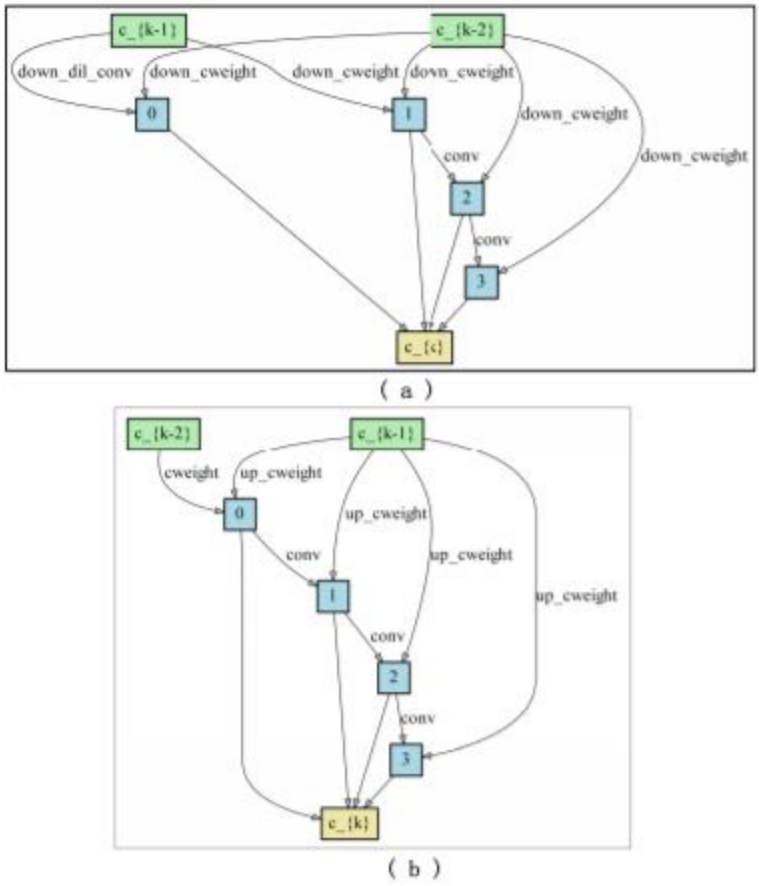
The plots below show how with an increase in training epochs, the accuracy (mIoU) increased and validation loss decreased. More epochs increases the accuracy of the model. However, it is important to not make the number of epochs too high. If the number of epochs is too high, it could result in overfitting. However, this graph is looking at test data accuracy so it is difficult to make conclusions about overfitting without training data accuracy.



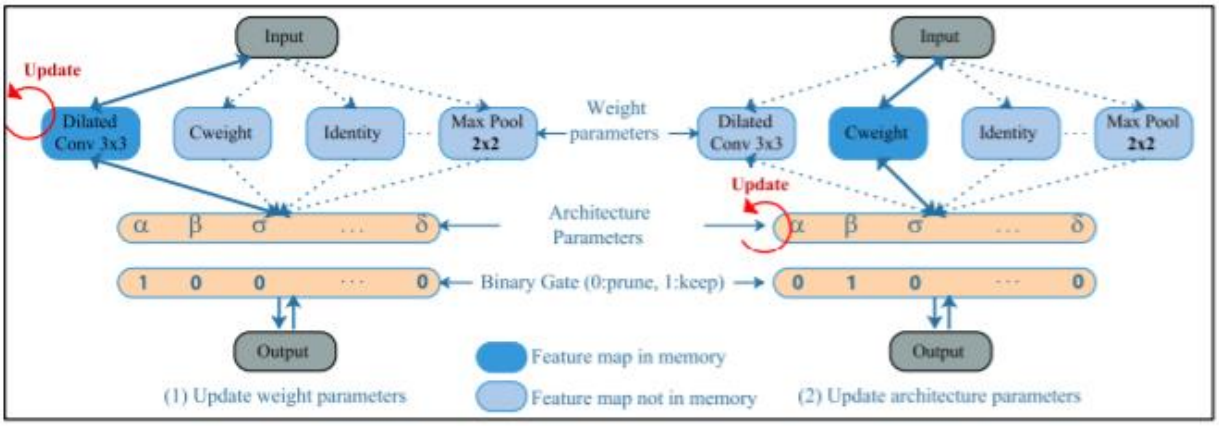
Paper Conclusions and Relevance to our Project:

The paper successfully showed that for each image type they looked at, their NAS-Unet performed the best, as seen by the high mIoU values. The DSC (Dice Similarity Coefficient) shows the similarity between all of the outputs. The high DSC values for NAS-Unet suggest that this model is the most consistent out of the architectures tested. One downside to their structure is that it takes a lot more time to train than the U-Net. This train time is the total days and hours when the batch size is 2. Another downside of the NAS-Unet is that it has a higher GM than the U-Net. This GM represents the GPU memory costs when the batch size is 2 also.

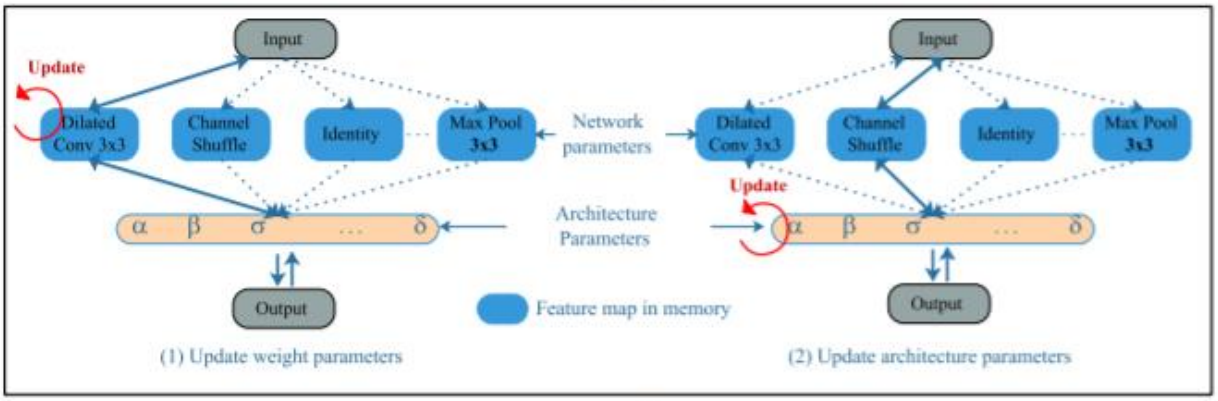
The final cell architectures they used in their NAS-Unet are depicted below. Here (a) represents the down scaling cells and (b) represents the up scaling cells. These cell architectures were found using their search strategies.



The high training time of a NAS-Unet is not very problematic for our project because once the model has been trained, predicting for each patient will be very quick. Thus, training time is not as important as the high accuracy benefits from the NAS-Unet. An issue is they had access to much more data. In order to mitigate this, techniques like cross validation could be used. There are also some minor modifications to their NAS-Unet that would need to be made to adapt it to our project. First of all, the primitive operations in the cells might need to be altered so the model is better suited for ankles in the bones instead of organs or nerves. This could be done with trial and error or the method they used which was the use of a memory saving search algorithm to find the best cell architecture. There is also chance the overall structure would need to be altered but this is unlikely as the paper had such high accuracies with their data. The memory saving search algorithms they tried were DARTS and ProxylessNAS. The difference between these were DARTS updated all paths when updating weight parameters and ProxylessNAS only updated one path when updating weight parameters. These strategies are depicted in the image below where (a) represents ProxylessNAS and (b) represents DARTS.



(a)



(b)

Another modification that needs to be made is the final output of the neural networks. Based on how the manual segmentations and validation framework is structured, a likelihood matrix is required for each bone. Thus, the output must be multiple matrices for each bone of interest instead of just one matrix. Each of these matrixes will be a binary matrix with 1 signifying if that pixel is likely part of the bone of that matrix and 0 otherwise. Another minor change that we would make is when reporting statistics, we would report both testing and training data accuracy. This way, it will be very clear if our model is under or overfitting. This will allow us to tweak the parameters and end at an optimized model.

Advantages of Paper:

The best aspect of this paper was that they utilized multiple types of images. Though none of the images are from the exact C-arm used in this project, by validating that their model worked with multiple image types shows that it is likely that it is a good architecture that could be used for our project. They also showed a method for finding the best cell architecture. This is the method they used and it is what could be used in this project. This will allow us to slightly modify their architecture so it is better suited to bones in an ankle C-arm image. Another major benefit of this paper is all of their code is open source. Thus, there would be no intellectual property issues with using this architecture. It gives us the freedom to not only use their research but build upon what they did so that we can find an architecture that better suits our problem. The paper also did a good job of using the established U-Net and FC-densenet structures as baselines. This allowed them to clearly prove that their NAS-Unet was a

better architecture than the U-Net and FC-densenet (which are established architectures) when trained and tested on the same data.

Disadvantages of Paper:

It would have been better if they had a more diverse data set. They did look at organs and nerves. However, looking at more anatomical features, for example bones, could have showed how their architecture could be applied to even more types of segmentations involving other types of anatomical features. Another major disadvantage of this paper is they used different sized sets for training in each of the experiments. This makes it difficult to draw conclusions across each of the experiments. This is because the results would be different if they all had the same number of training images. This could have given insight into which architectures were better for which types of images. Though these conclusions would not have been very valid even if the number of training images were the same (this is because there is a lot of variation in neural network training like the quality of the images and other factors that cannot be controlled) more insight would have been gained. Another thing this paper could have done is reporting training accuracy. This would have helped make overfitting much easier to avoid (if it occurred) and it would have conveyed it better to the readers.