Virtual Incision

Applied Internship - Machine learning

Type du Service

Nom du Service

Prof. Réf.	:Jay Carlson
Auteurs	: Gauthier Bassereau
Version	: 1 R 0
Date	: 30/09/2024
Promotion	: "N° Promotion"
Entreprise	: Entreprise
Période	: du…/../…. au ../../….

Cartouche de validation

Actions	Persons	Signatures	Dates
Redaction	Gauthier Bassereau	Signature :	
Validation	Saisir le nom du correcteur	Signature :	
Diffusion	Saisir le nom du professeur référent/tuteur	Signature :	

Acknowledgements

I would like to extend my heartfelt thanks to the entire Virtual Incision organization for their incredibly warm welcome to a young French stranger. From the moment I arrived, I felt supported and valued as part of the team.

I want to specifically thank my colleagues in the R&D team. Evan, your incredible support in helping me integrate my model with Holoscan was invaluable, and you were always available whenever I needed guidance. Nixon, working alongside you was a fantastic learning experience; we both grew together in our respective projects, and it was a pleasure to have you around. Jake, thank you for introducing me to the real America, and for your companionship during my time here

Ryan, our long discussions on AI and your willingness to share your knowledge meant the world to me. Noah and Fabrice, our time together—filled with laughter, late nights, and great conversations—will always be cherished. Taylor, thank you for your insightful talks and for helping me connect my project to the backend of MIRA.

A very special thanks goes to Jay. Your expertise and passion were always clear, but what I value most is how you became a true mentor to me. You taught me that nothing is out of reach, and you constantly pushed me to be more confident and better in everything I did. Thank you for your support, your belief in me, and of course, for all the laughs along the way.

Table of contents

A. Abstract

During my three-month internship at Virtual Incision, I focused on enhancing the visual and control capabilities of the MIRA robotic surgery system using machine learning methods. My work was divided into three main projects: image upscaling, image segmentation, and autonomous control.

For image upscaling, I utilized deep learning models such as EDSR and ESRGAN to increase the MIRA camera feed resolution from 1080p to 4K. While I achieved significant improvements in image quality, meeting the 50ms latency requirement for real-time surgical applications proved challenging.

In the image segmentation project, I implemented a framework using Label Studio and Meta's Segment Anything Model (SAM) for efficiently labeling and segmenting images from the MIRA camera feed. This setup facilitated the creation of a U-Net-based segmentation model that demonstrated promising results in identifying robotic arms and surgical tools.

The most interesting but also complex task involved automating the control of the MIRA robot through imitation learning. By implementing the Action Chunking Transformer (ACT) model and training it on needle manipulation demonstrations, I was able to create a proof of concept for AI-driven automation of robotic surgery tasks.

The company expressed strong interest in the project, the developed AI features could play a crucial role in the next-generation MIRA console. Although a proof of concept was created, the project requires a huge amount of further development. Future improvements will focus on leveraging foundation models and autoencoders to enhance performance, providing more robust models.

Overall, the internship provided me with invaluable experience in applying machine learning to real-world challenges in robotic surgery and successfully explored the potential of AI features for Virtual Incision's technology.

B. Introduction

1. Company Overview

Virtual Incision Corporation, founded in 2006 as a spin-off from the University of Nebraska, is a leading medical device company based in Lincoln, Nebraska. Specializing in robotic-assisted surgical technologies, Virtual Incision's core mission is to democratize access to minimally invasive surgery by developing compact, portable, and affordable robotic systems. Their flagship product, the MIRA (Miniaturized In Vivo Robotic Assistant) Surgical Platform, is designed to assist surgeons in performing complex abdominal surgeries.

Unlike traditional large-scale robotic systems that require specialized operating rooms, MIRA is lightweight, portable, and does not depend on a dedicated infrastructure, making it accessible to a broader range of healthcare facilities, from large hospitals to smaller surgery centers. MIRA is directly controlled by the surgeon, ensuring precise manipulation during surgery, while its advanced tools and high-quality cameras improve visibility and accuracy. Virtual Incision's mission is to make robotic surgery more affordable and accessible globally, revolutionizing the field of minimally invasive surgery.

Since its inception, Virtual Incision has achieved several major milestones, including securing over 200 patents and attracting significant investment. Notably, in January 2020, the company completed a Series B+ financing round, raising $20 million, which brought its total funding to over $100 million. In February 2024, Virtual Incision received FDA approval for colon surgeries using MIRA, a critical milestone that solidifies the company's place in the surgical robotics market. Additionally, Virtual Incision has partnered with NASA to test the MIRA system in space, further demonstrating the robot's potential in extreme environments.

Figure 1: ***MIRA Robot***

2. Research & Development

Virtual Incision is actively developing the next-generation MIRA console, scheduled for launch in 2026. The ongoing research and development efforts are focused on revamping the entire system architecture, from rewriting the backend and frontend to introducing entirely new hardware components. In addition, the company is expanding the use of AI in surgical robotics. By integrating advanced machine learning frameworks for image processing and control automation, Virtual Incision aims to push the boundaries of what's possible in robotic-assisted surgery. The new console is expected to offer improved precision, faster processing times, and enhanced automation features, further reducing the workload on surgeons and boosting surgical efficiency.

Following a recent tech conference, Virtual Incision and Nvidia have announced a partnership to implement Nvidia's Holoscan SDK, a groundbreaking project set to launch at the end of 2024. Nvidia's Holoscan SDK is a full-stack infrastructure that leverages GPU-accelerated computing to process real-time data streams with high efficiency. The next-generation MIRA console will harness this technology to build an end-to-end image processing pipeline entirely based on GPUs. This approach offers unmatched video latency, while also enabling faster post-processing algorithms, such as machine learning models, to run with minimal delay. By keeping the entire processing pipeline on the GPU, Virtual

Incision will be the first robotic medical company to integrate such cutting-edge technology, setting unmatched performance.

One the exciting goal of the next-generation console is to integrate AI-driven solutions that will eventually enable the automation of simple and repetitive tasks, and progressively improve to more complex task like suturing.

3. Internship Overview and Contribution to the New Console

As part of the efforts to advance the capabilities of the MIRA system, my internship at Virtual Incision focused on implementing and experiencing with machine learning enhancements that will be integrated into the new console. During my internship, I was tasked with improving the visual quality of the MIRA camera feed, creating a segmentation model for robotic tools and organs, and developing a demonstration of an AI autonomous task.

Upscaling and Image Enhancement - One of the major challenges with the current MIRA system is its 1080p resolution camera, which, while highly praised for its quality, does not fully utilize the capabilities of the 4K screen on the console. My mission was to explore the potential of using deep learning models, such as EDSR and ESRGAN, to upscale the 1080p camera feed to 4K resolution while meeting the real-time latency requirement of under 50 milliseconds.

Image Segmentation - I established a data-labeling pipeline using open-source tools like Label Studio and integrated Meta's Segment Anything Model (SAM) to create a framework for labeling and segmenting images from the MIRA camera feed. This work laid the foundation for developing a segmentation model capable of accurately detecting robotic tools, surgical instruments, and organs, crucial for enhancing the visual feedback provided to surgeons.

Autonomous Controls - In the most complex aspect of my internship, I worked on automating control of the MIRA robot using imitation learning, particularly for tasks like needle manipulation. I implemented the Action Chunking Transformer (ACT) model, demonstrating how AI can automate repetitive tasks such as grasping and passing a needle between robotic arms. This proof of concept is directly aligned with the future direction of the MIRA platform, as Virtual Incision seeks to integrate more automation into the new console, allowing the system to handle repetitive surgical tasks with greater precision and efficiency.

By contributing to these three areas—upscaling, segmentation, and autonomous controls—I have helped pave the way for AI integration into Virtual Incision's next-generation surgical platform, enhancing both the visual and control systems of the MIRA robot.

C. Missions

1. Upscaling and Image Enhancement

1.1. Objective

Virtual Incision has developed its own full HD camera, small enough to be integrated into the MIRA system (Figure 1). It is the world's smallest full HD camera (1080x1920 pixels), and it has garnered significant praise from surgeons in robotic-assisted surgery for its image quality when compared to competitors. The image provided by this camera is critical, as it is the primary feedback tool for the surgeon operating the MIRA robot. Despite this, the camera's 1080p resolution does not fully leverage the capabilities of the 4K screen used in the current console.

My first mission was to explore whether it was possible to apply machine learning to upscale the 1080p camera feed to 4K resolution, all while maintaining a latency of under 50 milliseconds.

1.2. Approach and Research

The first phase of this mission involved conducting a thorough literature review, primarily leveraging arXiv.org, which hosts many of the most recent advancements in deep learning. Image upscaling has been a well-researched topic for many years. Earlier techniques, such as linear and bicubic interpolation, were commonly used in the early 2000s. However, the emergence of deep learning in 2014, particularly Convolutional Neural Networks (CNNs), transformed the field by significantly improving image quality and scalability.

CNNs are at the core of modern image processing techniques. A convolution operation involves sliding a small matrix, or filter, over the input image to generate feature maps. This process enables the network to detect patterns such as edges, textures, and shapes, which are crucial for tasks like super-resolution. CNNs rely on learning filter parameters during training to identify relevant patterns based on the specific task. For super-resolution, CNNs extract deep features from a low-resolution input, gradually reducing its spatial dimensions while increasing the number of channels to capture complex hierarchical features. These features are then reconstructed into a higher-resolution image through upsampling techniques like transposed convolutions or sub-pixel convolution layers. Residual connections, introduced by the ResNet architecture in 2015, further improved CNNs by allowing deeper networks to be trained without suffering from vanishing gradients, leading to finer detail preservation during image reconstruction.

Training upscaling models is relatively straightforward and doesn't require specialized datasets. A large, varied set of images is sufficient for training. The process involves downscaling the images and applying augmentations before feeding them into the model. The model then computes the error between the upscaled output and the original image. To optimize efficiency, it's common to train on smaller patches of images, such as 128x128, rather than the full high-resolution images. This approach not only conserves memory but also enables faster convergence and thus training, as fewer pixels need to be processed per iteration. Additionally, smaller patches help the model focus on learning local features such as edges, textures, and details, which are key elements in upscaling, without having to process the entire image at once. By doing so, the model effectively reconstructs fine details when applied to larger images during inference.

See Figure 2 for a visual understanding of such upscaling models architecture.

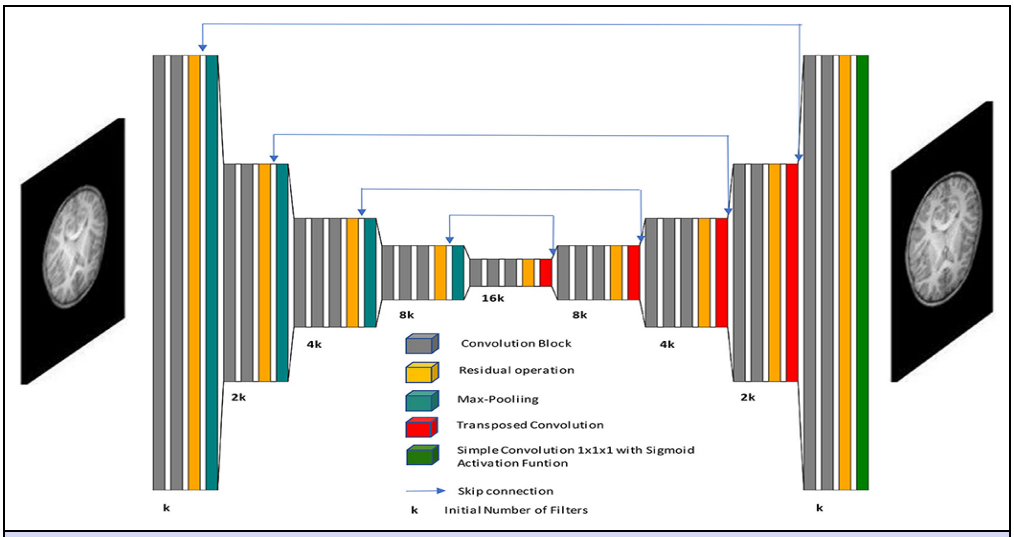

Figure 2: ***Basic Model for Super Resolution using CNNs and Residual connections***

In 2016, Generative Adversarial Networks (GANs) took the field a step further by introducing a new paradigm for image super-resolution. GANs employ two neural networks: a generator that attempts to create high-resolution images, and a discriminator that aims to distinguish between real and generated images. This type of training is called adversarial training, the two models are trained against each other, it enables the generator to become progressively better at producing images with sharper details and more realistic textures compared to traditional CNN-based methods. Both the generator and the discriminator in GAN-based models are specialized CNN architectures, with the generator focusing on upscaling and the discriminator on evaluating image realism.

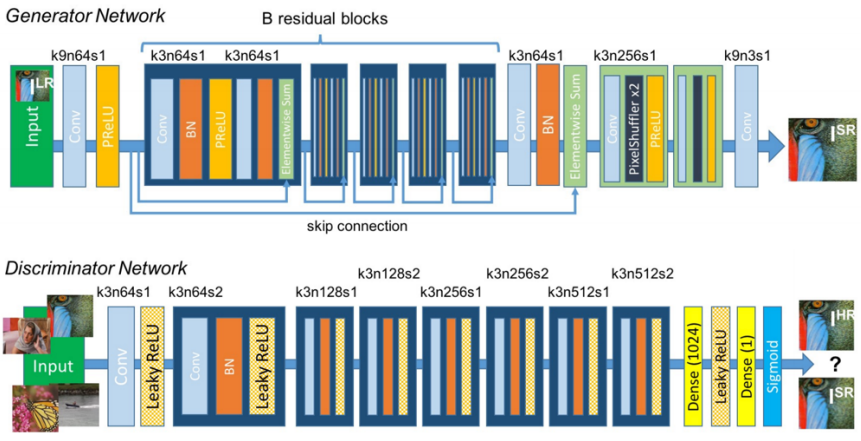

Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

Figure 3: ***SRGAN, Generative model for Super Resolution***

1.3. Implementation

After thorough research, I decided to start by implementing the EDSR (Enhanced Deep Residual Networks), one of the most recent CNN architectures for upscaling without generative methods. Later, I explored more advanced methods like ESRGAN (Enhanced Super-Resolution GAN), state-of-the-art generative models for image super-resolution.

To maintain consistency across all machine learning projects, our team agreed to use PyTorch as the primary framework for implementation. Although I had prior experience with TensorFlow, I was fully on board with this decision as it is the library used by almost all published research. This project provided a valuable opportunity for me to become proficient with PyTorch, which is getting increasingly popular in the field.

Dataset - To train the models, I used two widely recognized datasets, Div2K and Flick2K, from Kaggle, which provided over 3,000 high-resolution images. To handle the large dataset size during training, I created a custom PyTorch dataset instance that dynamically loads and augments images. For each training step, a random image is selected, rotated, flipped, and cropped into 128x128 patches. This patch-based approach is widely used in training upscaling models, as it significantly improves memory efficiency and accelerates convergence.

Model Development: Implementing the EDSR model was a straightforward yet educational process, as it provided hands-on experience with PyTorch and convolutional networks. The model consists of several convolution layers, activation functions, and a PixelShuffle layer for upsampling. Most importantly, I have been able to measure the impact of residual connections in such large models.

Residual connections, introduced by the ResNet architecture, play a critical role in addressing the vanishing gradient problem that often occurs when training deep networks. In deep learning models, gradients can become very small as they are backpropagated through the layers, leading to slow learning or even stagnation. Residual connections mitigate this by allowing the model to "skip" one or more layers, passing information directly from earlier layers to later ones. This direct path not only improves gradient flow during training but also helps the model capture finer details by combining both shallow and deep feature representations.

In the context of image super-resolution, these connections are particularly crucial because they enable the network to better preserve and reconstruct high-frequency details such as textures and edges, which are often lost in traditional deep networks. This was evident in my experiments, where models with residual connections consistently outperformed those without, both in terms of convergence speed and the quality of the upscaled images. As a result, residual connections allowed the EDSR model to achieve sharper, more detailed results compared to earlier methods.

After successfully implementing and understanding the EDSR model, I moved on to ESRGAN (Enhanced Super-Resolution GAN), a more advanced model for image super-resolution. While ESRGAN builds upon the same fundamental principles as EDSR, including the use of convolutional layers and residual connections, it introduces a deeper architecture with additional layers to capture more complex features. However, the key difference lies in how ESRGAN is trained.

1.4. Training and Optimization

Training the EDSR model was relatively straightforward. I employed a basic L1 loss function, which measures the pixel-wise difference between the generated and ground-truth images. This loss function is commonly used in image upscaling tasks as it directly minimizes the error across the entire image, producing results that align closely with the original image. Given that EDSR is a non-generative model, the focus was on optimizing the model's ability to produce clean and sharp upscaled images without introducing artifacts.

I trained the model using patches of 128x128 from high-resolution images to conserve memory and improve convergence speed. Each training batch involved random cropping, flipping, and rotating these patches, which helped the model generalize better to different types of textures and patterns. After a few hours of training, I could already see noticeable improvements in image clarity and detail preservation. One advantage of the EDSR model is its relatively low computational demand during training, making it quicker to experiment with different hyperparameters.

Moving on to ESRGAN, the training process was significantly more complex due to its adversarial nature and the introduction of multiple loss functions. ESRGAN uses a combination of three distinct losses:

Adversarial Loss - This loss is computed using binary cross-entropy to measure how well the discriminator can distinguish between real and generated images. The generator's objective is to produce images that can “fool” the discriminator into thinking they are real.

Perceptual Loss - This loss is calculated using the feature maps from a pretrained VGG network. Instead of simply comparing pixel values, perceptual loss compares high-level features, ensuring that the generated images preserve important structures and details as seen by human perception, such as textures, edges, and object shapes.

Content Loss - This loss ensures that the generated image maintains the overall structure and content of the original image. While perceptual loss focuses on finer details, content loss helps maintain the overall accuracy of the upscaled image in relation to the original.

Training such generative model requires careful balancing of the generator and discriminator. If one becomes too strong relative to the other, the training can collapse, resulting in poor image quality. To avoid this, I regularly monitored the loss values using Tensorboard, adjusting hyperparameters such as learning rates, batch sizes, and the balance between the three losses as needed.

A key challenge in ESRGAN training is preventing the discriminator from overpowering the generator. This balancing act required frequent tuning to maintain a stable adversarial game between the two models. ESRGAN's perceptual and adversarial losses make the training more sensitive to hyperparameter changes, requiring much more experimentation compared to EDSR. For instance, adjusting the learning rates for both the generator and the discriminator was crucial to prevent one model from converging too quickly or slowly relative to the other.

Despite these challenges, ESRGAN produced significantly sharper and more realistic images, especially in regions with fine textures and details. The combination of perceptual loss and adversarial training enabled ESRGAN to generate upscaled images that were perceptually much closer to the ground-truth high-resolution images compared to EDSR.

In summary, while EDSR provided good initial results with relatively simple training, ESRGAN pushed the boundaries of image realism by introducing a more sophisticated training framework, although with increased complexity and training time. Both models were optimized for real-time performance by first converting the model to the common ONNX format and then compile to TensorRT, which is a Nvidia format that enable even faster inference on their GPUs.

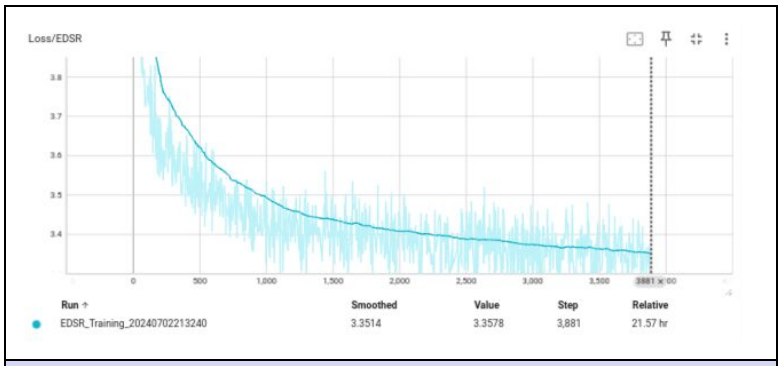

Figure 4: ***ESDR Training loss***

Figure 4 shows the training loss of the ESDR model.

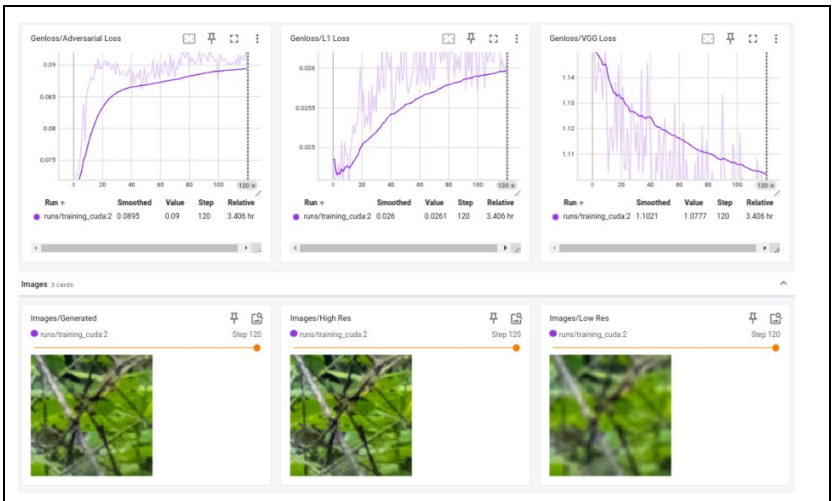

Figure 5: ***ESRGAN Training loss + Sampled Patches***

Figure 5 shows the combined losses and sampled patches, it is interesting to notice how the training losses don't have the same behavior as usual trainings because of its adversarial nature.

1.5. Results and Challenges

Despite achieving visually impressive results, the biggest challenge remained meeting the strict 50ms latency requirement. Initially, the unoptimized model exhibited a latency of around 400ms, which I managed to reduce to 200ms by compiling the model to TensorRT format. However, further reducing latency while maintaining image quality proved difficult. A notable issue with GANs, in particular, is their tendency to produce artifacts. This often occurs when the adversarial loss becomes too dominant, causing the generator to focus on fooling the discriminator rather than faithfully reconstructing high-resolution details. As a result, GAN-generated images can sometimes display unnatural patterns or distortions, which became especially apparent when attempting to apply ESRGAN to highly compressed images from the MIRA camera feed.

As the project progressed, it became clear that further downsizing the models to meet the real-time performance requirement significantly degraded image quality. The more I reduced model complexity, the less noticeable the improvements in upscaled image quality became, making the benefits marginal and not worth the extensive effort required to push the project forward. The work-to-reward ratio was simply not favorable, as the reduced model did not justify the effort needed to put it into production. After extensive optimization attempts and careful consideration, we ultimately decided that the project wasn't worth pursuing further. The downsized models failed to deliver significant improvements in image quality, and the work-to-reward ratio was simply not justifiable. The effort required to optimize the model for real-time performance while maintaining high image quality outweighed the benefits, especially when compared to other projects that offered a better return on investment for the company.

That said, the project still provided me with invaluable learning opportunities. It deepened my understanding of deep learning, particularly in training and fine-tuning GANs, and taught me how to balance competing objectives like image quality and latency. Moreover, the experience of integrating these models into the Holoscan SDK was essential for future machine learning projects.

Figure 6 shows a zoomed image from the validation set using ESDR model.

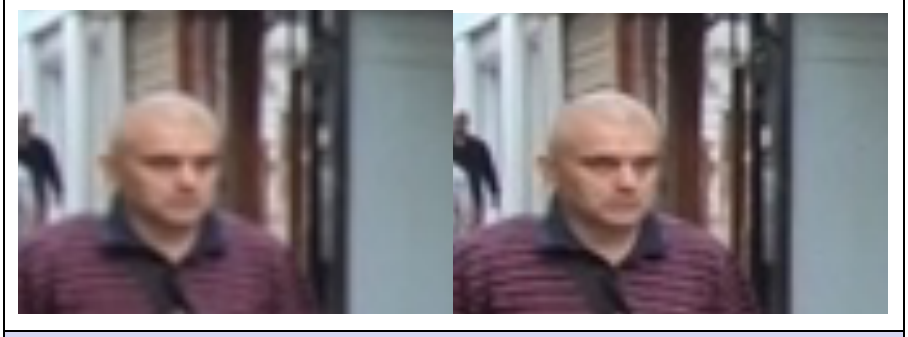

Figure 6: ***Left, Bicubic Upscaling – Right, ESDR Upscaling***

Figure 7 shows a zoomed image from the validation set using ESRGAN model.

Figure 7: ***Left, Bicubic Upscaling – Right, GAN Upscaling***

2. Image segmentation

2.1. Objective

This project aimed to establish a working pipeline for labeling and segmenting data from the MIRA camera feed. The primary goal was to create a framework that would allow anyone within the company to label data and build a machine learning model capable of accurately detecting surgical tools, robotic arms, and organs. Segmentation could play a crucial role in organ detection and identifying robotic arms, where different color filters could be applied to differentiate the arms from the patient's body. By darkening the robot arms, which are of lesser interest to the surgeon, we could improve visual clarity. Additionally, we considered varying the darkness of the robotic arms to indicate depth, potentially addressing the lack of 3D perception with a single RGB sensor. The overarching goal was to build a scalable framework for segmentation and test it with a basic model.

2.2. Implementation

As always, the first step involved researching the available tools for data labeling. While several online tools like Roboflow offered labeling solutions, they came at a cost and didn't provide enough value for our specific needs. I turned to open-source alternatives and found that Label Studio was the most comprehensive option for our task.

Given that the project required mask labeling, I needed a tool that could incorporate pre-trained models, especially those designed for segmentation. The Segment Anything Model (SAM) from Meta was an ideal candidate, as it significantly speeds up labeling by providing high-quality segmentations out of the box.

To enable company-wide access to this labeling tool, I implemented Label Studio on the company's server. This allowed anyone within the network to access the tool and start labeling data. I also integrated the SAM model into the setup using Docker, running it on one of the server's GPUs. While this integration was a challenging task, the result was a tremendous boost in labeling efficiency. Setting up Docker, managing Ubuntu servers, and ensuring proper network access were all new to me, but I gained valuable experience through this process. Though it took some time and a few tough days, the effort paid off, and the system was fully functional.

Once the labeling infrastructure was in place, my colleague Nixon and I began labeling the data and developing a basic segmentation model to test the pipeline. We chose a U-Net architecture for the model, a well-established choice for segmentation tasks. Despite having limited labeled data, our initial results were promising (see Figure 9), thanks to the high-quality data that Virtual Incision had previously recorded in its labs.

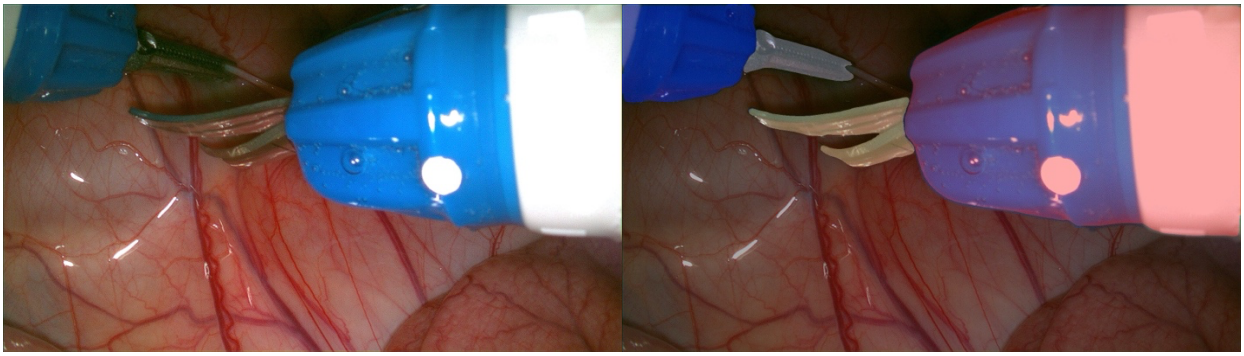

Figure 8: ***Segmentation of Left Arm, Left Tool, Right Arm, Right Tool***

3. Autonomous Controls

3.1. Objective

This project was the most challenging, time-consuming, and exciting task I worked on during my internship. While my previous missions focused on visuals, this one aimed to explore how AI could be used to control the MIRA robot. One of the main challenges with MIRA's kinematics is the lack of wrist joints, which makes tasks like needle manipulation particularly difficult. The objective of my mission was to demonstrate that it is possible to automate some simple tasks using AI. MIRA is equipped with hardware capable of supporting AI, and my goal was to enhance its control system by showcasing AI-driven manipulation skills, specifically with a needle.

3.2. Approach and Research

To begin, I spent several days researching the state-of-the-art methods in robotic automation. This is a rapidly evolving field, with a wealth of research papers being published every year. In robotics, there are two primary approaches to automation: supervised learning and unsupervised learning.

Supervised learning, particularly imitation learning, involves training a model on expert demonstrations, allowing the robot to mimic skilled human operators. Unsupervised learning, including reinforcement learning (RL), allows the robot to learn by interacting with its environment without the need for labeled training data. While RL has historically been the go-to method for robotic control, it is computationally expensive and complex to implement. Recently, imitation learning has emerged as a promising alternative, especially with the development of transformer-based architectures, offering efficient learning and scalability.

Reinforcement learning excels in scenarios requiring optimization or planning where intermediate steps are not defined, making it ideal for end-to-end tasks. However, in my case, the objective was not to perform entire surgical procedures but to automate a single, repetitive task in varying environments. For this reason, I chose to focus on imitation learning, which is more suited to the problem at hand.

Imitation learning is a relatively straightforward approach: I recorded myself performing needle manipulations in different orientations, collecting around 100 demonstrations. The task then involved training an AI model to predict, based on a single image, the next 100 joint values of MIRA.

One key challenge for the model is to not fall into cumulative error – when a model only predicts the next step, we can easily see how each prediction get some kind of inevitable errors. This means that the prediction is probably a bit more out of distribution than the previous one, meaning that when the model predicts the next step, it will have even more error due, at a certain time, the robot will end up somewhere completely out of distribution and then the model has no chance to recover once out of distribution, it will be in a configuration it has never seen in the dataset and will get only worse and worse predictions. Multiple papers have tried to answer this problematic and one of the answers to the problem is to predict a sequence of actions and not only the next, meaning that the whole next sequence will have some kind of error, but it will be constant over all the steps and not cumulative.

Another key challenge was accounting for multimodality — the existence of multiple valid ways to achieve the same result. For example, when navigating around an obstacle, there might be several correct trajectories, but a naive model might average these paths and end up hitting the obstacle. Handling this issue is critical for achieving robust control in real-world applications.

3.3. Implementation

A competitor to Virtual Incision, Intuitive, had published a paper using imitation learning to automate similar tasks with their robotic system. They employed the Action Chunking Transformer (ACT) model, which I decided to recreate for my project.

Dataset - I recorded demonstrations of myself picking up and passing a needle from one arm of the robot to the other, each lasting about 20 seconds. In total, I gathered around 100 demonstrations. As in previous projects, I created a custom PyTorch dataset, but this time, I applied mean and standard deviation for normalization. I also had to ensure that each demonstration was properly segmented and padded to avoid mixing them during training. One major challenge was synchronizing the visual data with MIRA's joint values, as the frames had been recorded without timestamps. With the help of my colleague Evan, who wrote a custom operator, I was able to sync the data properly. I also wrote a script to extract only the relevant segments from the recorded videos, ensuring a clean and organized dataset. Constructing this dataset was one of the most intricate aspects of the project due to the numerous small but crucial details that needed to be accounted for.

Codebase - Recreating the ACT model was a significant task. It was my first time implementing such a large model and working with transformers, which took me several days to fully understand and code. The model had about 10 million parameters, similar to the ESRGAN model, but with a much more complex structure due to its multiple components. I consulted several open-source implementations to guide my coding. The ACT model also incorporates a Variational Encoder to address the multimodality issue I previously described. This was my first experience implementing this type of architecture, and it was both challenging and rewarding. Additionally, I made a point to build a scalable codebase, which I learned from my previous projects that it was a critical need. I designed the project so that components like models, datasets, and training parameters could be easily modified or replaced. The main objective of my codebase was that each training trial was entirely saved, from the model configuration, the detailed logging, model checkpoints, data distribution, ensuring that the codebase was clean and maintainable.

Training - Training the model was relatively straightforward, though I encountered a major issue early on due to an overlooked bug in my loss function. This led to some confusing initial results, but once corrected, training proceeded smoothly. Compared to previous models, the training times were significantly longer. I experimented extensively with different hyperparameters, normalization techniques, and prediction methods, testing various configurations to optimize the model's performance.

3.4. Results

After numerous trials and adjustments, I successfully demonstrated a machine learning model that could control the MIRA robot autonomously, completing a needle manipulation task. This video demonstration served as proof that AI can be effectively applied to automate control tasks with MIRA, laying the groundwork for future developments in robotic automation on MIRA.

Figure 9: ***Demonstration of picking and handing off a needle / Speeded up***

D. Critical Analysis and Discussion

1. Reflection

During my internship at Virtual Incision, I had the opportunity to work on challenging and meaningful projects that significantly contributed to my growth as an engineer. I was tasked with improving both the visual and control aspects of the MIRA robotic system using cutting-edge machine learning techniques, and these experiences allowed me to apply and expand my technical knowledge in real-world applications.

My work spanned three key areas—image upscaling, segmentation, and autonomous control. In the image upscaling project, I successfully implemented and trained deep learning models like EDSR and ESRGAN, achieving noticeable improvements in image quality from the MIRA camera feed. This was my first experience working with GAN-based architectures, which significantly expanded my understanding of generative models and their application to image enhancement. Although I was able to upscale the images effectively, meeting the stringent latency requirements for real-time surgery proved challenging. I optimized the model by converting it to TensorRT format and reducing the latency to 200ms, but this was still above the 50ms target. Despite not fully solving the latency issue, I gained valuable experience in optimizing deep learning models for real-time applications and balancing model complexity with performance.

My image segmentation project was equally rewarding. Establishing the data labeling pipeline using Label Studio and integrating it with Meta's Segment Anything Model (SAM) was a significant achievement. It not only streamlined the data labeling process but also laid the foundation for future segmentation tasks at the company. The experience of managing Ubuntu servers and configuring Docker containers for GPU use was a technical challenge that I overcame, enhancing my ability to handle infrastructure alongside model development. By choosing U-Net for the initial segmentation model, I was able to demonstrate promising results, particularly in detecting robotic arms and surgical tools. Despite the limited amount of labeled data, the high-quality data already recorded by Virtual Incision allowed the project to progress successfully.

The most technically demanding and exciting project was the autonomous control task, where I recreated the Action Chunking Transformer (ACT) model to automate needle manipulation using imitation learning. This was my first time implementing a large-scale transformer model, and I gained deep insights into model architectures, sequence prediction, and multimodality handling. I also became adept at creating custom datasets and synchronizing data, crucial for training robotic control systems. While training the model took longer than expected and required extensive hyperparameter tuning, the final result—demonstrating AI-based control of the MIRA robot—was a significant achievement. This proof of concept showed the potential of AI in automating precise tasks, laying the groundwork for future robotic automation projects.

Each project presented its own set of challenges, from reducing the latency of upscaling models to synchronizing visual and joint data for the robot. I faced difficulties in balancing the performance of generative models with real-time constraints, managing large datasets, and coding complex architectures like transformers. However, each challenge taught me valuable lessons, particularly in terms of model optimization, working with large-scale systems, and building scalable, modular codebases that can be extended in future projects. My ability to learn and adapt quickly, coupled with strong collaboration with my colleagues, allowed me to tackle these challenges and contribute meaningfully to the company's goals.

2. What's next ?

The future of this project lies in the development of a foundation model capable of performing multiple surgical tasks, such as grabbing, cutting, and knot-tying, within a unified framework. The idea is to move away from creating separate models for each task and instead build a single, versatile model that can execute all necessary robotic actions. The beauty of this approach is that such a model will be able to learn from each task, meaning that as the model improves at one task (like grasping a needle), it will leverage that knowledge to enhance its performance on other tasks (like cutting or knot-tying). This

approach is based on the principle of shared learning, where knowledge from one task can benefit others, leading to more efficient and adaptable learning.

A crucial part of improving the performance of this foundation model lies in the image encoder. In my current implementation, I used a ResNet-based encoder pretrained on ImageNet, but this approach is limiting when applied to surgical environments. The next step will involve developing a self-supervised image encoder that can extract rich representations specific to the robotic surgery context. This encoder will be trained on MIRA's data, allowing it to understand the nuances of the surgical environment.

Self-Supervised Training: A self-supervised autoencoder could be used to encode the visual inputs into a useful latent vector that captures essential features such as depth, segmentation, surface normals, and could even be points clouds, the more the better. By training the encoder to predict these features from the raw camera feed, the model will develop a far more nuanced understanding of the environment than a standard ResNet trained on unrelated data like ImageNet.

Rich Feature Extraction: This new encoder would go beyond simple classification tasks to build a comprehensive understanding of the scene. The ability to extract multiple types of information (segmentation, depth, normals, etc.) from a single image means that the encoder will produce a much richer latent space. This, in turn, will make downstream tasks (like controlling the robot for grasping or cutting) much more efficient and accurate.

Another critical area for improvement is the control policy. Currently, the control policy relies on imitation learning to predict actions based on human demonstrations. While effective, the diffusion policy represents a more advanced method for generating sequences of actions that are more reliable and robust, particularly because of its natural multi-modality representation.

Reinforcement learning should also play a crucial role later in the process, first by optimizing the policy learned by imitation learning, making it more efficient and more robust. In a second time, reinforcement learning could be the solution to make the foundation model continuously learn from its experience.

In summary, the future direction of this project will involve developing a multitask foundation model capable of handling all surgical tasks, leveraging a self-supervised image encoder tailored to the robotic surgery domain, implementing advanced diffusion policies to optimize control, and finally optimizing and getting continuous improvements on the model.

Before leaving Virtual Incision, I had the opportunity to share my vision of the project to Ryan and Nicholas, which will take over my project, and have done some preliminary work. Below is a pretrained model from META that extracts a depth map from single RGB images, which can be crucial to use as training for the visual auto-encoder I was explaining above.

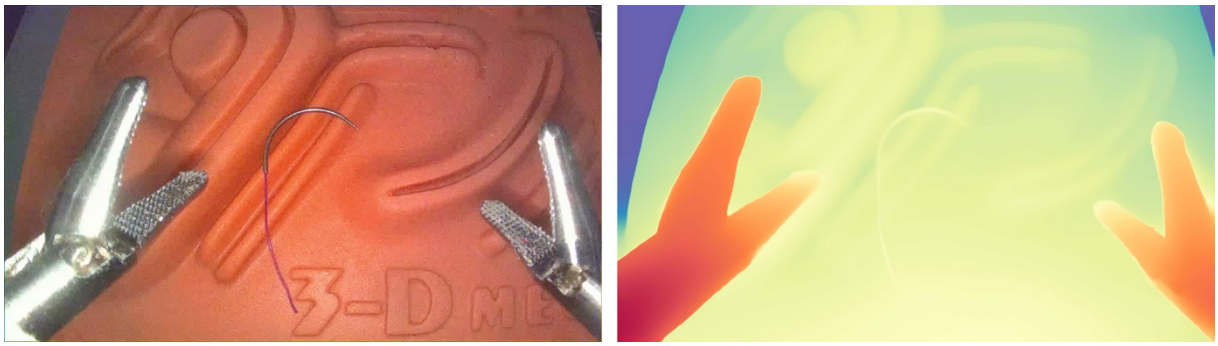

Figure 10: **ViT-Huge, Depth extraction**

E. Conclusion

F. List of references