CAMERA POSE ESTIMATION FROM SURGICAL VIDEOS WITH DEEP LEARNING

PROJECT N°7

STUDENTS : Andrea Naclerio
Ali Shadman Yazdi
Siavash Taleghani

TUTOR : Alberto Rota

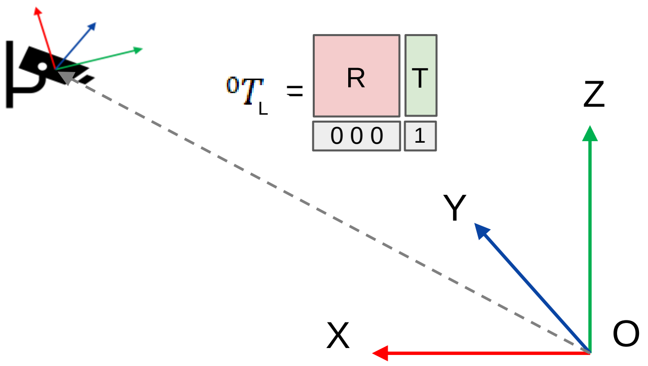
Camera Pose Estimation

What is camera pose estimation?

A homogeneous transformation matrix that describes position and orientation of a camera

Why do we need camera pose?

- 3D reconstruction
- Enhanced perception
- Organs and lesions localization



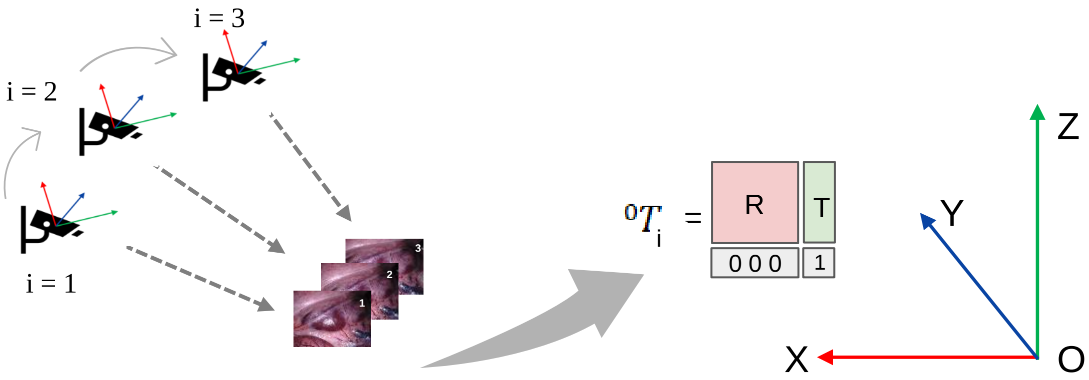
State of the art

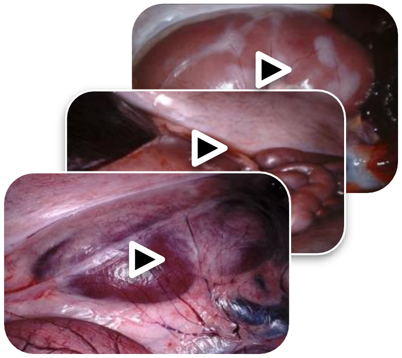
Visual SLAM

- Tracking points of interest
- 3D positioning through triangulation
- Developing 3D map
- Mainly used in car localization

Deep Learning

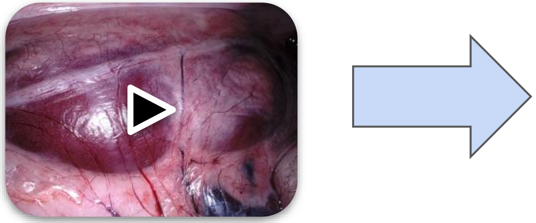
- Feature extraction through DCNN
- Camera pose and image information relationship establishment
- Predicting camera pose

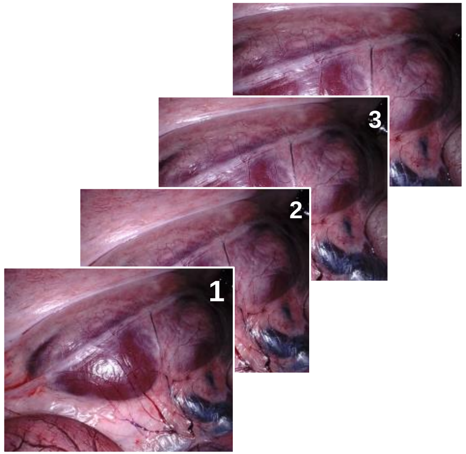




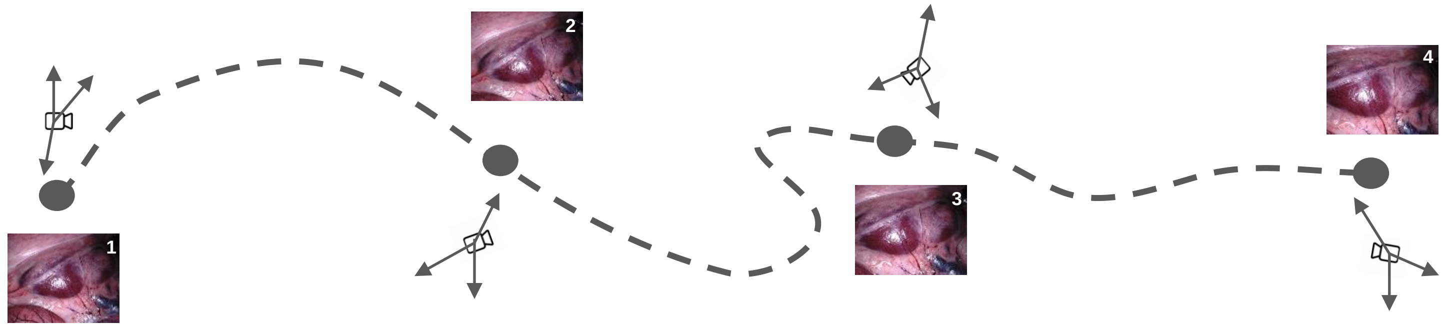
DATASET

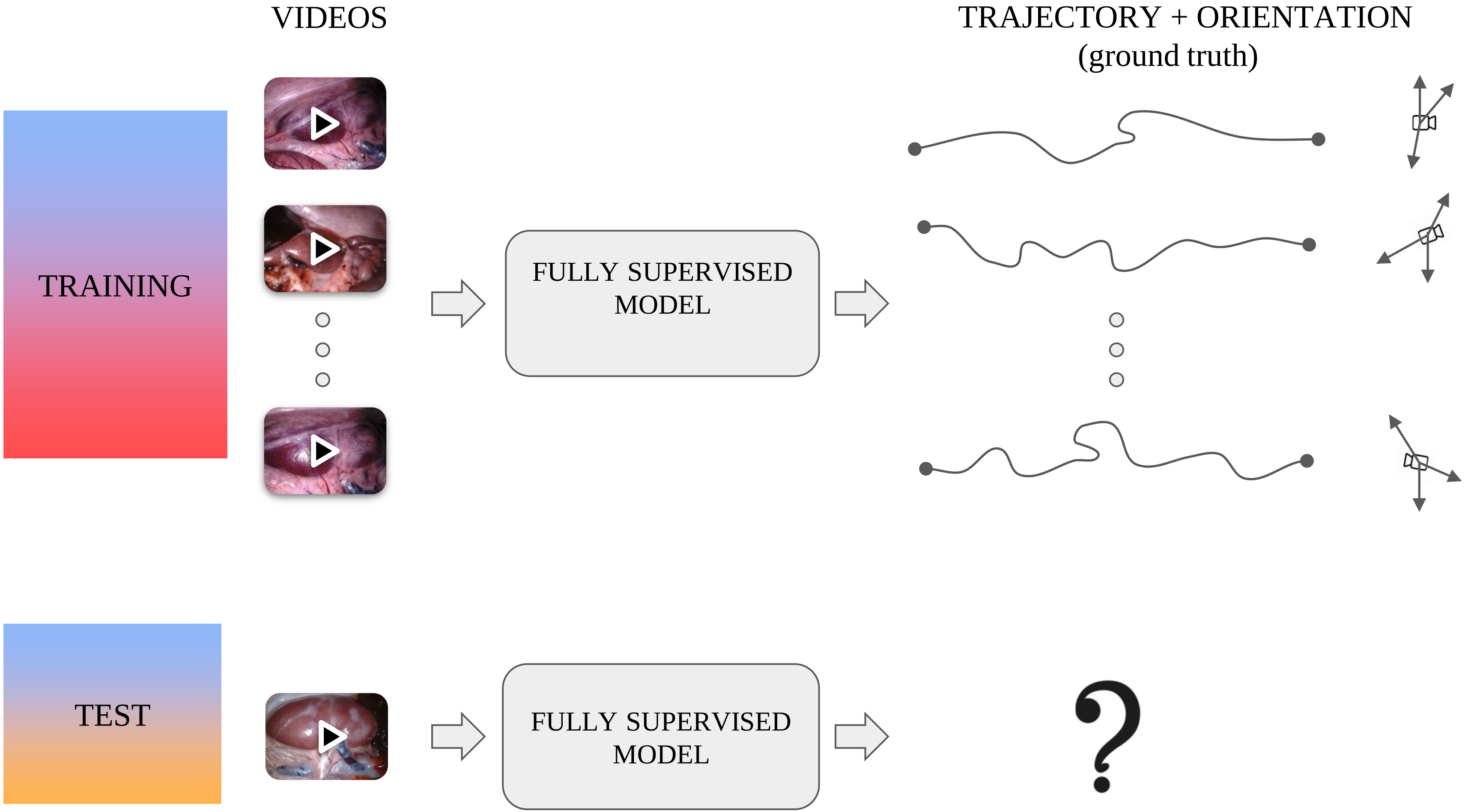
27 monocular videos annotated



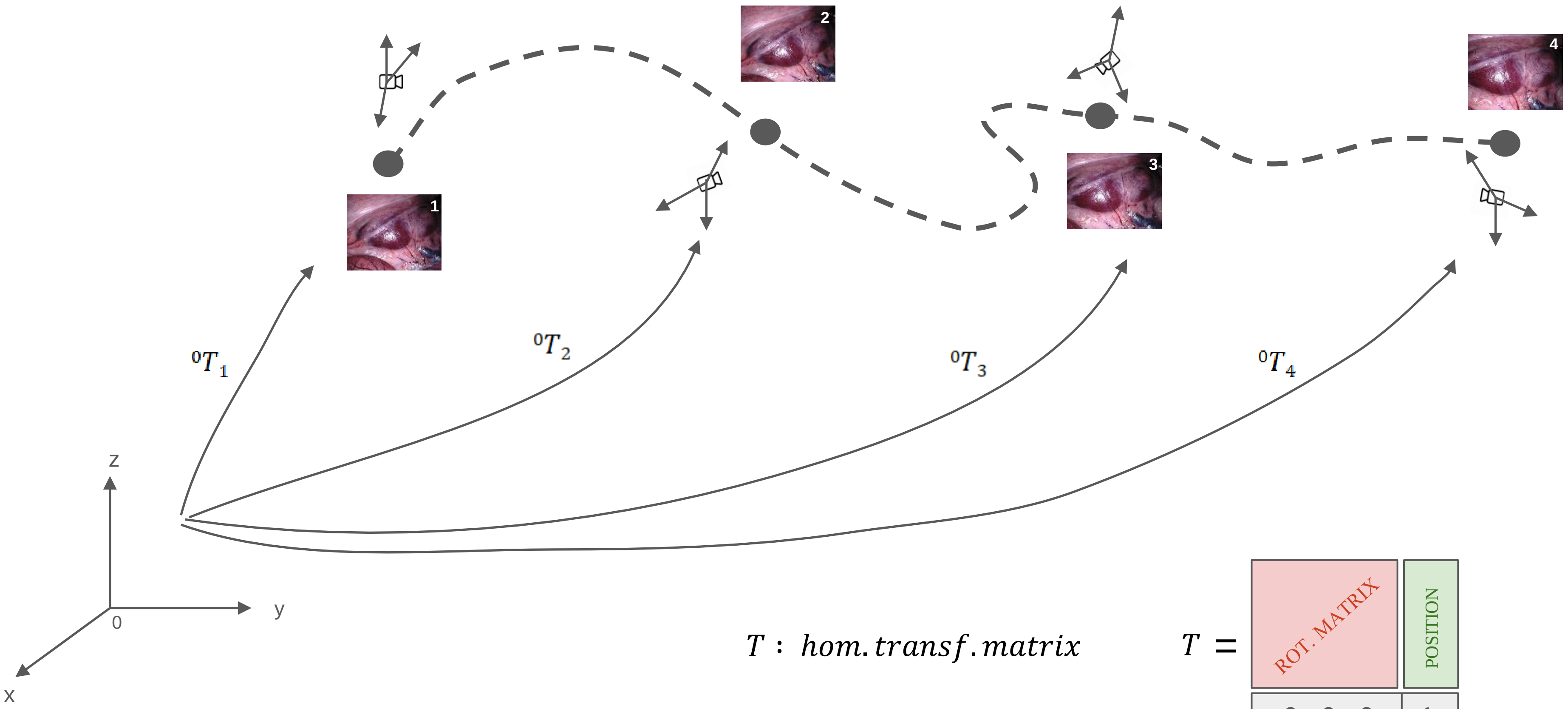


FRAMES




VIDEOS
TRAJECTORY + ORIENTATION
(ground truth)
TRAINING
FULLY SUPERVISED
MODEL
TEST
FULLY SUPERVISED
MODEL
?

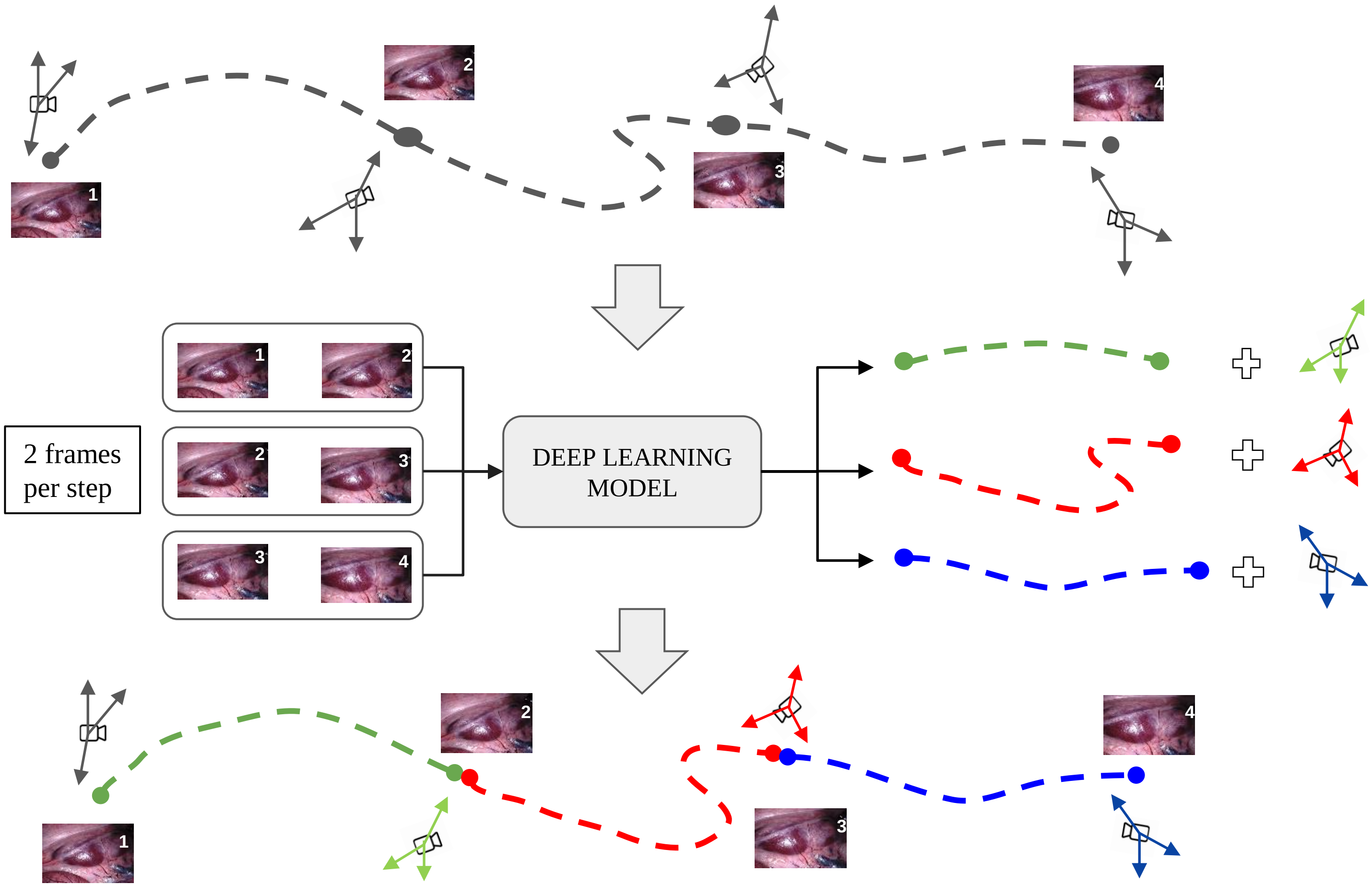
Ground-Truth data



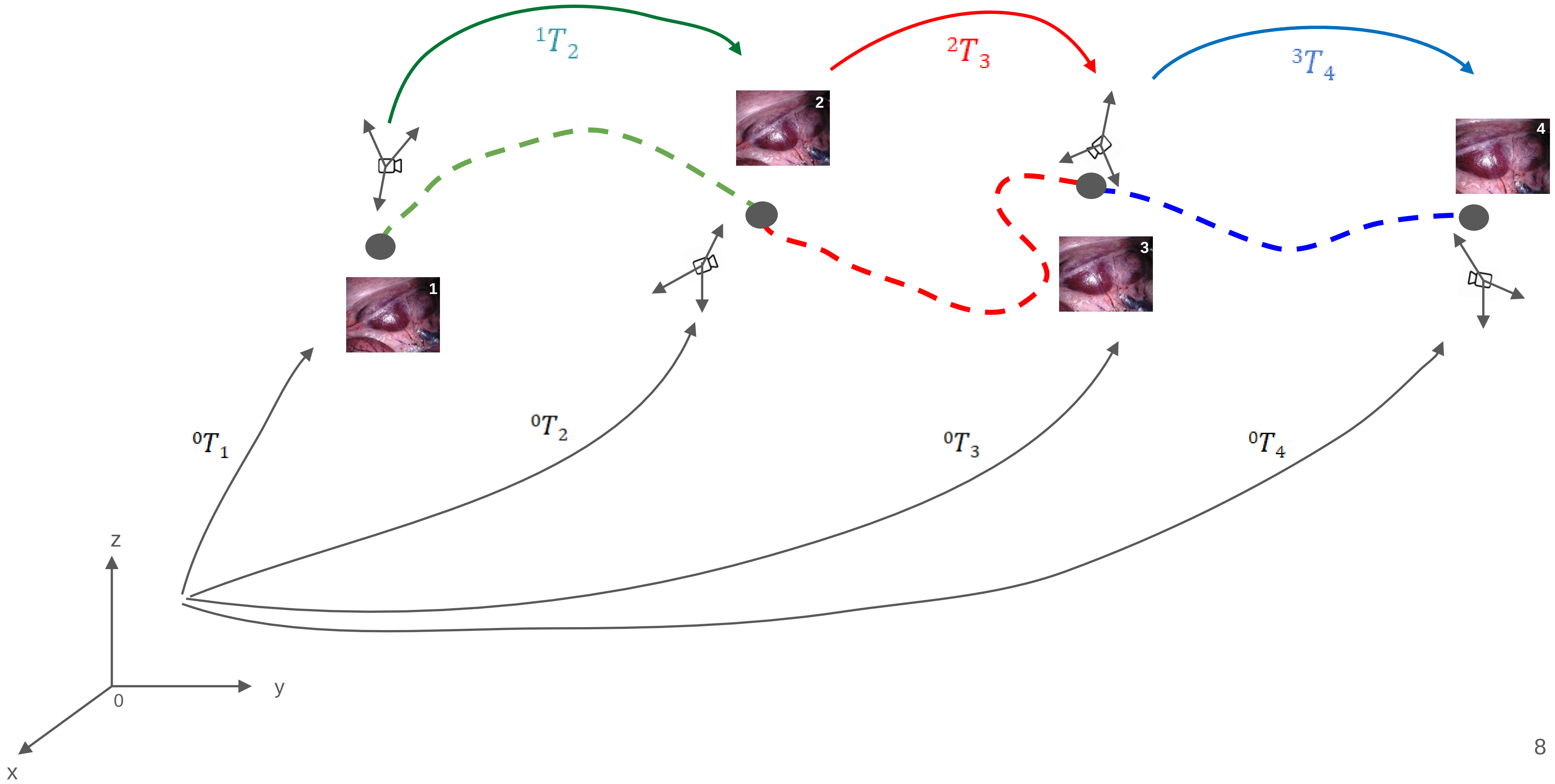
$T:\ hom.transf.matrix$

$$T = \begin{bmatrix} \text{ROT. MATRIX} & \text{POSITION} \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

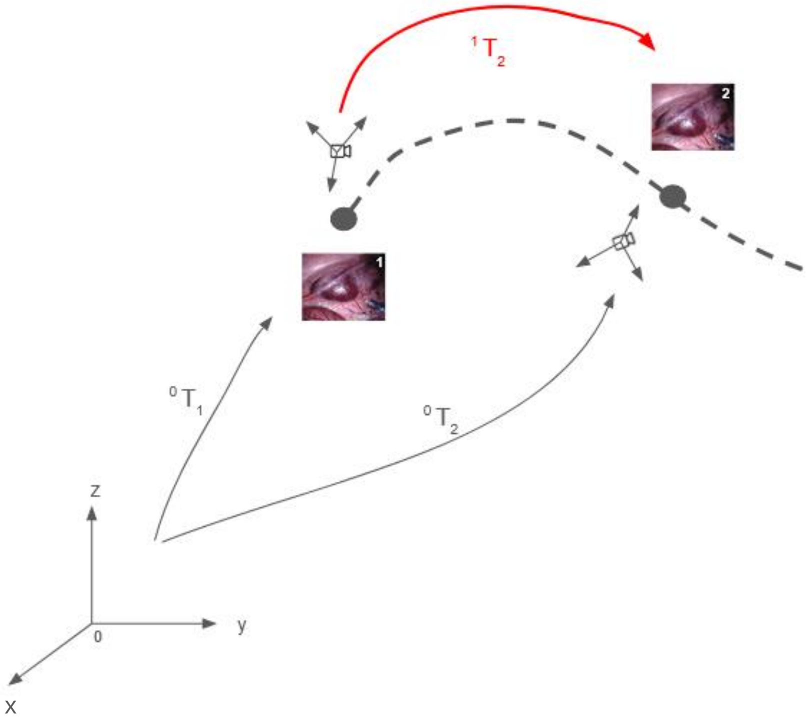
Problem decomposition



New transformation matrix



Output computation



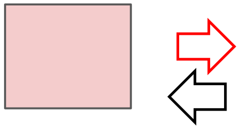
$$ {}^{1}T_{2} = {}^{0}T_{1}^{-1} * {}^{0}T_{2} $$

$$ {}^{1}T_{2} = \begin{bmatrix} \text{ROT. MATRIX} & \text{POSITION} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} $$

ROT. MATRIX ⇒ $[\, \alpha \,,\, \beta \,,\, \gamma \,]$

POSITION ⇒ $[\, x \,,\, y \,,\, z \,]$

INVERSE PROBLEM



$[\,\alpha\,,\,\beta\,,\,\gamma\,]$

ROTATION MATRIX *EULER ANGLES*

$$R_{x,y',z''}(a,\beta,\gamma) \;=\; \begin{bmatrix} \boldsymbol{c\beta\; c\gamma} & -\boldsymbol{c\beta\; s\gamma} & \boldsymbol{s\beta} \\ \boldsymbol{sa\; s\beta\; c\gamma + ca\; s\gamma} & -\boldsymbol{sa\; s\beta\; s\gamma + ca\; c\gamma} & -\boldsymbol{sa\; c\beta} \\ -\boldsymbol{s\beta\; ca\; c\gamma + sa\; s\gamma} & \boldsymbol{s\beta\; ca\; s\gamma + sa\; c\gamma} & \boldsymbol{ca\; c\beta} \end{bmatrix}$$

$$R \;=\; \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix}$$

$$\boldsymbol{\beta} = Atan2\,(\pm\sqrt{{r_{23}}^2 + \,{r_{33}}^2}, r_{13})$$

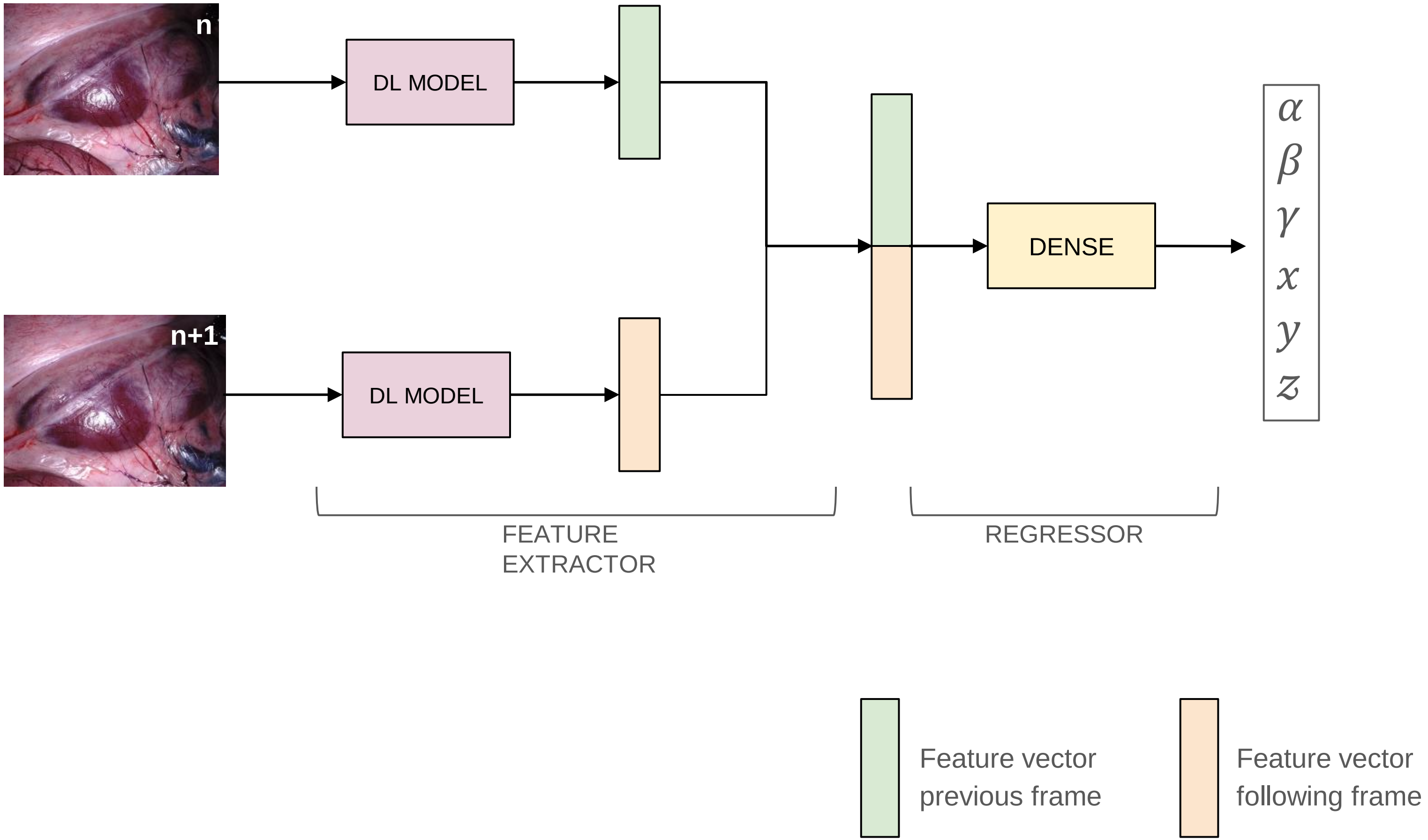
$$\boldsymbol{a} \;= Atan2\,(\frac{r_{33}}{c\boldsymbol{\beta}}, \frac{-r_{32}}{c\boldsymbol{\beta}})$$

$$\boldsymbol{\gamma} = Atan2\,(\frac{r_{11}}{c\boldsymbol{\beta}}, \frac{-r_{12}}{c\boldsymbol{\beta}})$$

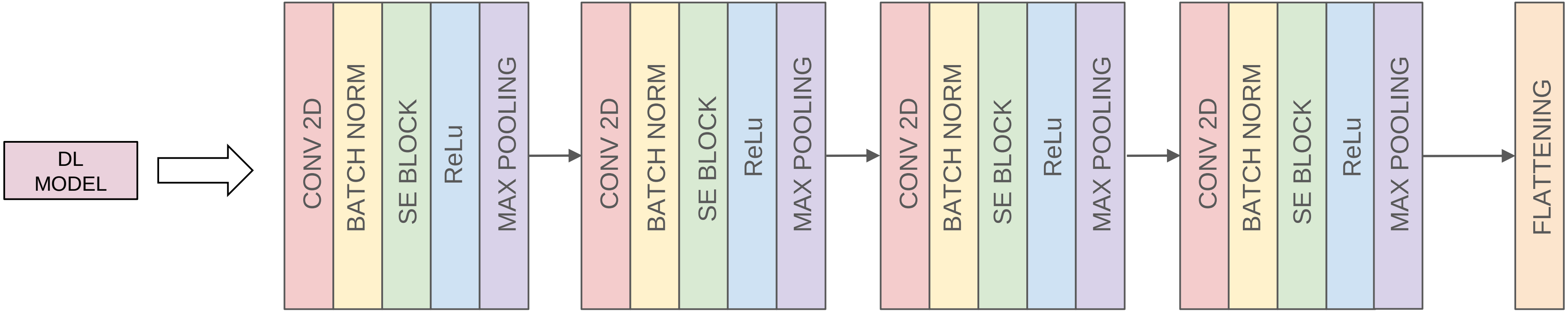
$$\Downarrow$$

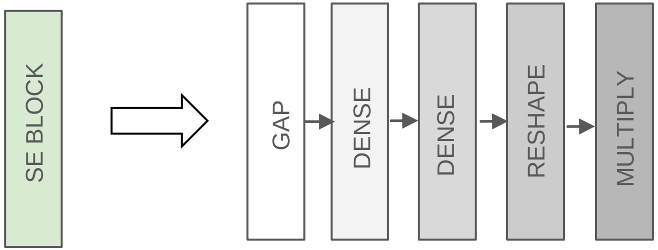
$$y = [\,\alpha \;\; \beta \;\; \gamma \;\mid\; x \;\; y \;\; z\,]$$

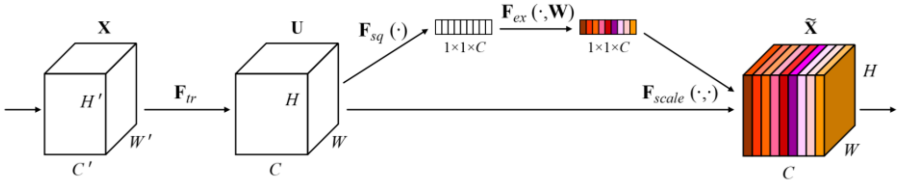
Two tails architecture



From scratch architecture





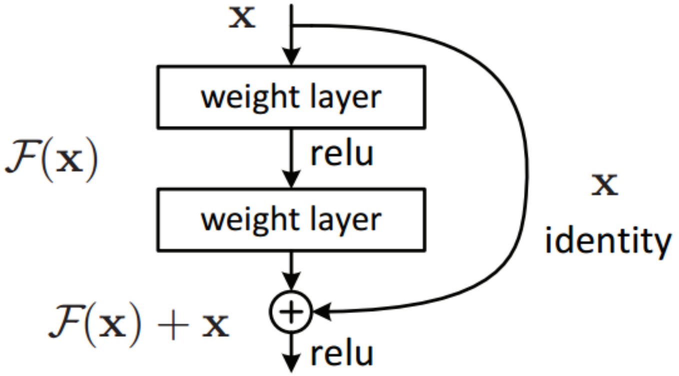


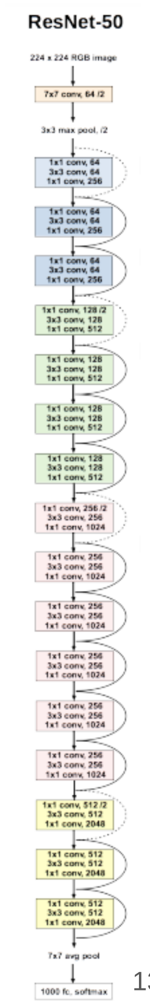
ResNet50 MODEL

- ResNet-50 introduced the concept of residual learning to address the challenge of training very deep neural networks and mitigating the vanishing gradient problem.
- A residual block contains a shortcut or skip connection that skips one or more layers, allowing the network to learn the residual (difference) between the input and output of those skipped layers.
- Weights were initialized to pre-trained model from image-net, but adjusted to fit our dataset during training

"Instead of figuring out the entire journey, let's figure out the changes we need to make at each step, and then just add those changes to the starting point.“

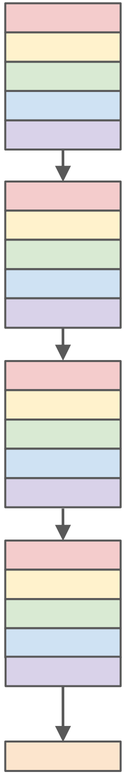
\- chat GPT et al.(2024)





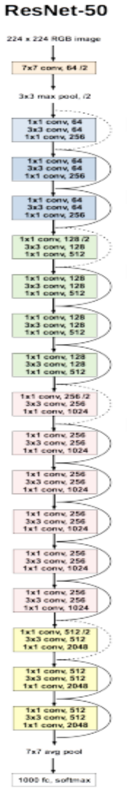
Model training

FROM SCRATCH MODEL



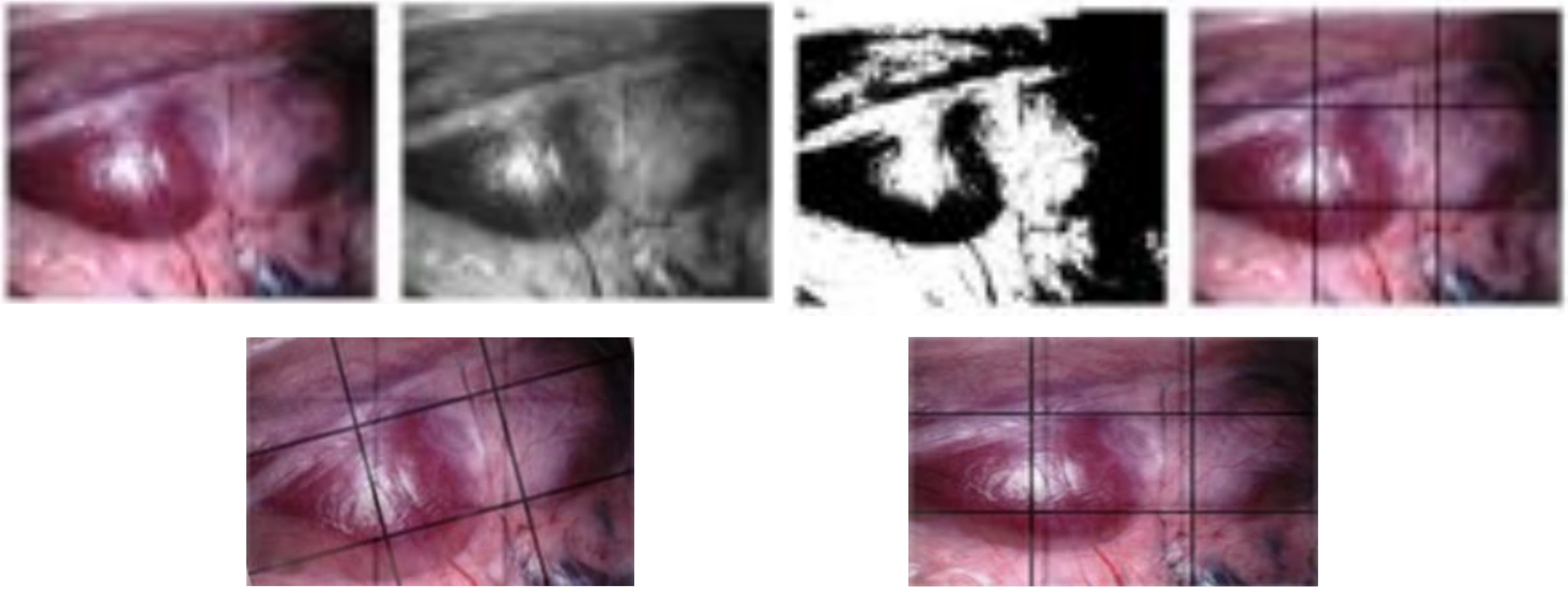
- TRAINING VIDEOS : **3 (v2, v3, v4)**
- NUM. TRAINIGN IMAGES **: 1784**
- TEST VIDEO: **1 (v1)**
- LOSS FUNCTION : **Mean Square Error (MSE**)
- BATCH : **12**
- LEARNING RATE (LR) : **0.0001**
- CALLBACKS:
  - EARLY STOPPING PATIENCE : **12**
  - REDUCE LR ON PLATEU FACTOR : **6**
- NUM. PARAMETERS : **40 millions**

RESNET50 MODEL



- TRAINING VIDEOS : **3 (v2, v3, v4)**
- NUM. TRAINIGN IMAGES **: 1784**
- TEST VIDEO: **1 (v1)**
- LOSS FUNCTION : **Mean Square Error (MSE**)
- BATCH : **4**
- LEARNING RATE (LR) : **0.0001**
- CALLBACKS:
  - EARLY STOPPING PATIENCE : **10**
  - REDUCE LR ON PLATEU FACTOR : **3**
- NUM. PARAMETERS : **340 millions**

Image preprocessing



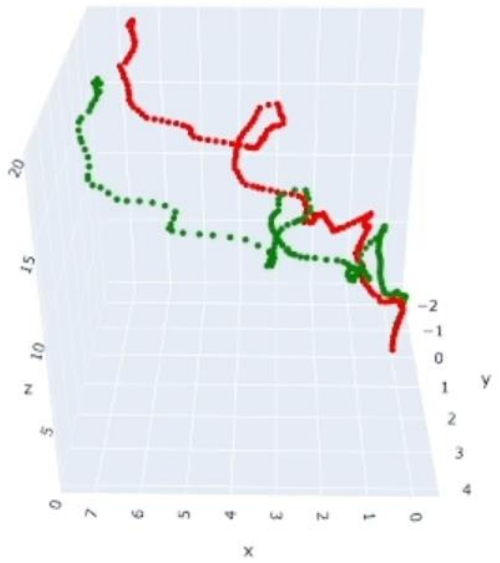
- RGB image → (m, 512, 640, 3) → 983,040 data points per image
- Binary image → (m, 512, 640, 1) → 327,680 data points per image

Effects of preprocessing

		Translation error	Rotation error
Base line		--	--
Grayscale		↑	↓
Binary		↑	↑
Grid		↑	↓

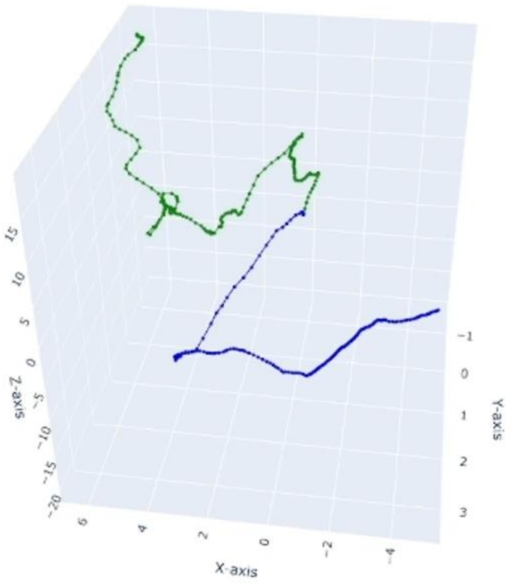
FINAL RESULTS

FROM SCRATCH MODEL



FROM SCRATCH MODEL VS TARGET TRAJECTORY

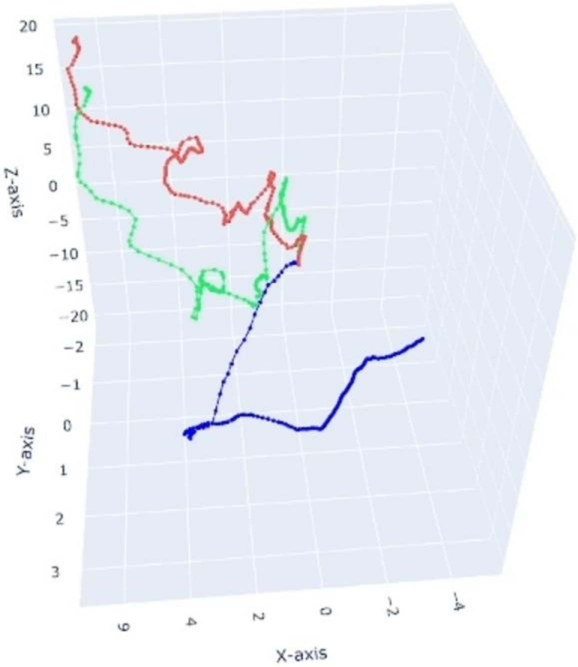
RESNET50 MODEL

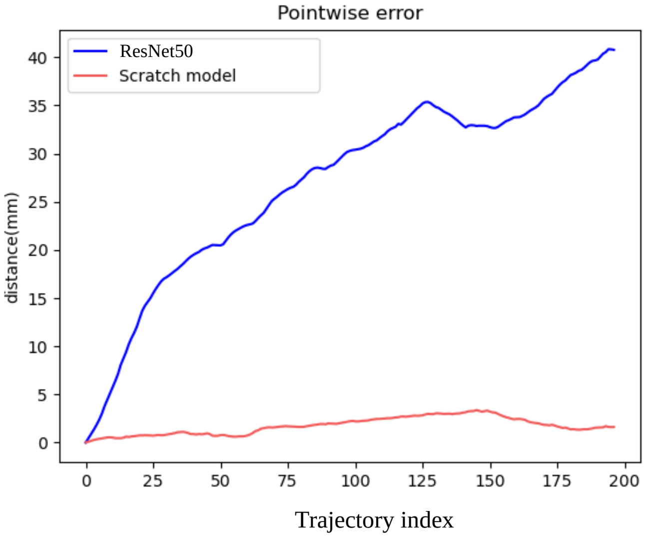


RESNET 50 VS TARGET TRAJECTORY

RESULTS COMPARISON

FROM SCRATCH MODEL VS RESNET 50 VS TARGET TRAJECTORY





NUMERICAL METRICS

	Scratch model	Resnet50
Mean Angular Error (degree)	10.7	12.4
Euclidian Distance (mm)	1.7	26.9

CONCLUSION

Limitation

- Texture-less tissues and Lambertian Reflection
- GPUs limitation
- Computationally intensive
- Overfitting

Future work

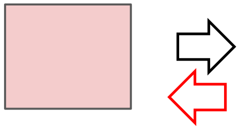
- Structural similarity
- Larger sequential
- Non sub-sequence frames
- Pretrained model on endoscopic camera dataset
- Ensembled light weight models

THANK YOU

FOR YOUR ATTENTION

QUESTIONS ?

DIRECT PROBLEM



$[\,\alpha\,,\,\beta\,,\,\gamma\,]$

ROTATION MATRIX *EULER ANGLES*

R : *rotation matrix*

$$R_{x,y',z''}(a,\beta,\gamma) = R_x * R_{y'} * R_{z''}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\boldsymbol{a} & -s\boldsymbol{a} \\ 0 & s\boldsymbol{a} & c\boldsymbol{a} \end{bmatrix} \qquad R_{y'} = \begin{bmatrix} c\boldsymbol{\beta} & 0 & s\boldsymbol{\beta} \\ 0 & 1 & 0 \\ -s\boldsymbol{\beta} & 0 & c\boldsymbol{\beta} \end{bmatrix} \qquad R_{z''} = \begin{bmatrix} c\boldsymbol{\gamma} & -s\boldsymbol{\gamma} & 0 \\ s\boldsymbol{\gamma} & c\boldsymbol{\gamma} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{x,y',z''}(a,\beta,\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\boldsymbol{a} & -s\boldsymbol{a} \\ 0 & s\boldsymbol{a} & c\boldsymbol{a} \end{bmatrix} * \begin{bmatrix} c\boldsymbol{\beta} & 0 & s\boldsymbol{\beta} \\ 0 & 1 & 0 \\ -s\boldsymbol{\beta} & 0 & c\boldsymbol{\beta} \end{bmatrix} * \begin{bmatrix} c\boldsymbol{\gamma} & -s\boldsymbol{\gamma} & 0 \\ s\boldsymbol{\gamma} & c\boldsymbol{\gamma} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{x,y',z''}(a,\beta,\gamma) = \begin{bmatrix} c\boldsymbol{\beta} & 0 & s\boldsymbol{\beta} \\ s\boldsymbol{a}\, s\boldsymbol{\beta} & c\boldsymbol{a} & -s\boldsymbol{a}\, c\boldsymbol{\beta} \\ -s\boldsymbol{\beta}\, c\boldsymbol{a} & s\boldsymbol{a} & c\boldsymbol{a}\, c\boldsymbol{\beta} \end{bmatrix} * \begin{bmatrix} c\boldsymbol{\gamma} & -s\boldsymbol{\gamma} & 0 \\ s\boldsymbol{\gamma} & c\boldsymbol{\gamma} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{x,y',z''}(a,\beta,\gamma) = \begin{bmatrix} c\boldsymbol{\beta}\, c\boldsymbol{\gamma} & -c\boldsymbol{\beta}\, s\boldsymbol{\gamma} & s\boldsymbol{\beta} \\ s\boldsymbol{a}\, s\boldsymbol{\beta}\, c\boldsymbol{\gamma} + c\boldsymbol{a}\, s\boldsymbol{\gamma} & -s\boldsymbol{a}\, s\boldsymbol{\beta}\, s\boldsymbol{\gamma} + c\boldsymbol{a}\, c\boldsymbol{\gamma} & -s\boldsymbol{a}\, c\boldsymbol{\beta} \\ -s\boldsymbol{\beta}\, c\boldsymbol{a}\, c\boldsymbol{\gamma} + s\boldsymbol{a}\, s\boldsymbol{\gamma} & s\boldsymbol{\beta}\, c\boldsymbol{a}\, s\boldsymbol{\gamma} + s\boldsymbol{a}\, c\boldsymbol{\gamma} & c\boldsymbol{a}\, c\boldsymbol{\beta} \end{bmatrix}$$