

WebApp results & Learning to Sort

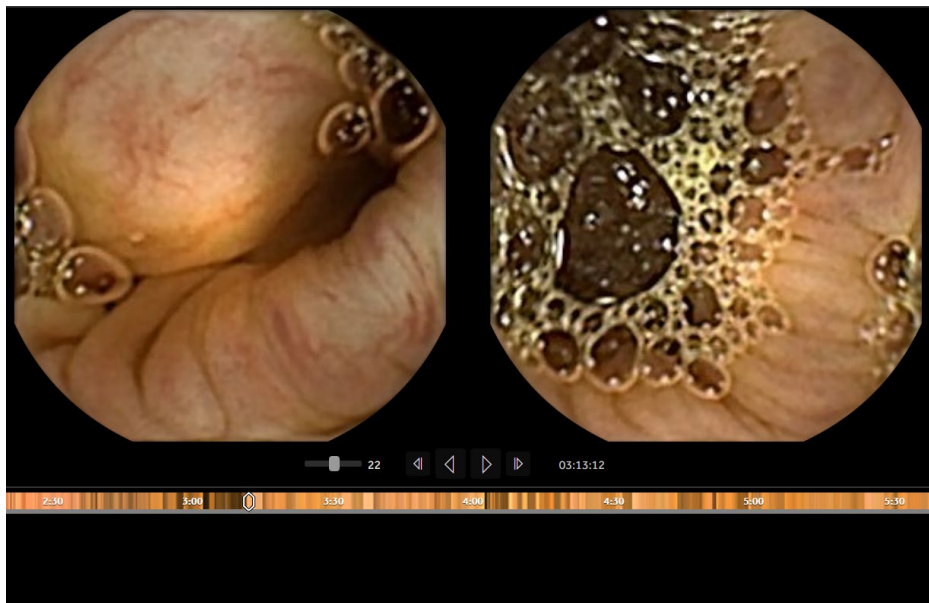
Pere Gilabert Roca
October, 2020

Index

1. Medical Experiments. WebApp results
2. Learning to Sort

1. Medical Experiments. WebApp results

The Problem



- Videos are over **8h long**
- Doctors spend a lot of time revising frames
- Videos are accelerated **to reduce reviewing time**
- Frontal and Back Camera are show at once
- Some polyps appear in just one frame

Rapid Software

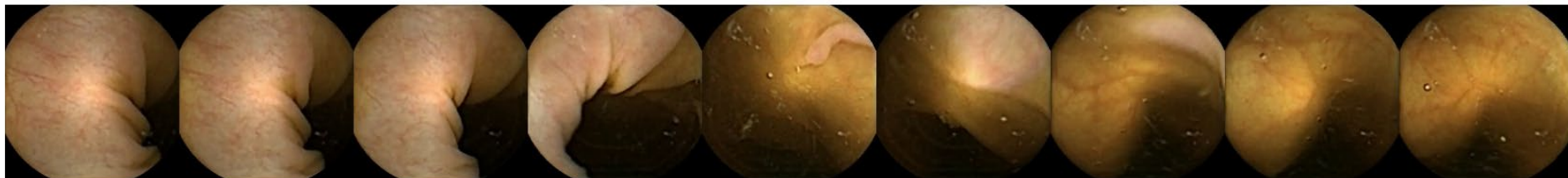
Our propose

- **WebApp** that helps physicians to visualize images with the highest score
- Give **context** by showing near frames

Elapsed time 00:00:05

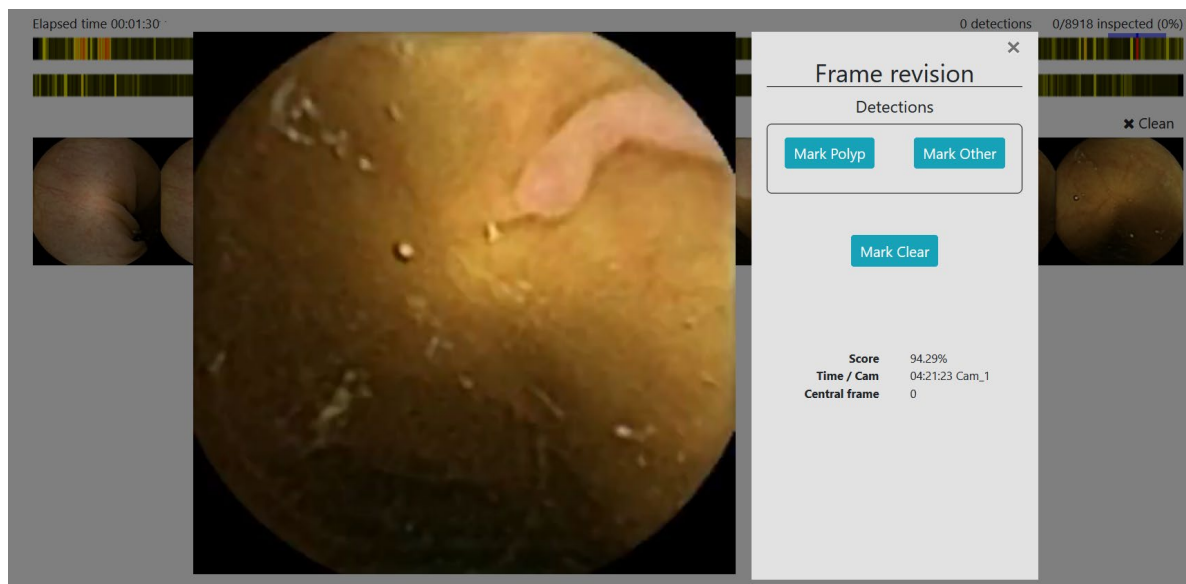
0 detections 0/8918 inspected (0%)

✕ Clean



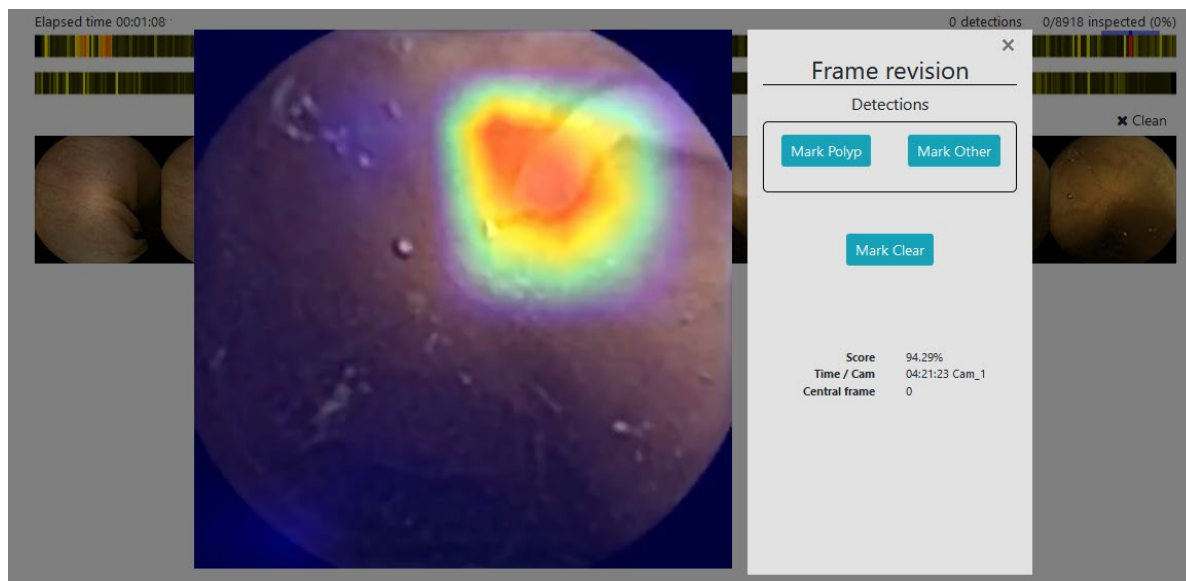
Our propose

- **WebApp** that helps physicians to visualize images with the highest score
- Give **context** by showing near frames
- Frames can be enlarged for a better visualization



Our propose

- **WebApp** that helps physicians to visualize images with the highest score
- Give **context** by showing near frames
- Frames can be enlarged for a better visualization



Experiments



3 Experts from CAPRI



3 Non-experts from Vall d'Hebron



22 Videos to revise



2 Tools used: **Rapid Software** & **WebApp**

Results

	000	002	007	023	031	039	040	062	066	067	084	085	088	096	101	114	024	052	077	090	095	117
B1(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B1(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
chiui1(R)	-	0/2	-	5/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	2/4	1/2	-	0/1	-	0/1	-
chiui1(W)	-	0/2	-	4/6	-	2/2	-	4/6	-	3/5	-	2/2	-	1/3	-	3/4	-	3/3	-	5/5	-	1/2
chiui2(R)	0/1	-	5/13	-	7/16	-	4/6	-	0/5	-	2/3	-	3/5	-	1/2	-	-	1/3	-	4/5	-	1/2
chiui2(W)	1/1	-	9/13	-	9/16	-	5/6	-	4/5	-	2/3	-	4/5	-	1/2	-	2/2	-	1/1	-	1/1	-
chiui3(R)	-	0/2	-	4/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	1/4	0/2	-	1/1	-	0/1	-
chiui3(W)	-	0/2	-	4/6	-	1/2	-	5/6	-	3/5	-	1/2	-	0/3	-	3/4	-	-	-	-	-	-

Results

	000	002	007	023	031	039	040	062	066	067	084	085	088	096	101	114	024	052	077	090	095	117
B1(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B1(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
chiui1(R)	-	0/2	-	5/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	2/4	1/2	-	0/1	-	0/1	-
chiui1(W)	-	0/2	-	4/6	-	2/2	-	4/6	-	3/5	-	2/2	-	1/3	-	3/4	-	3/3	-	5/5	-	1/2
chiui2(R)	0/1	-	5/13	-	7/16	-	4/6	-	0/5	-	2/3	-	3/5	-	1/2	-	-	1/3	-	4/5	-	1/2
chiui2(W)	1/1	-	9/13	-	9/16	-	5/6	-	4/5	-	2/3	-	4/5	-	1/2	-	2/2	-	1/1	-	1/1	-
chiui3(R)	-	0/2	-	4/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	1/4	0/2	-	1/1	-	0/1	-
chiui3(W)	-	0/2	-	4/6	-	1/2	-	5/6	-	3/5	-	1/2	-	0/3	-	3/4	-	-	-	-	-	-

Results

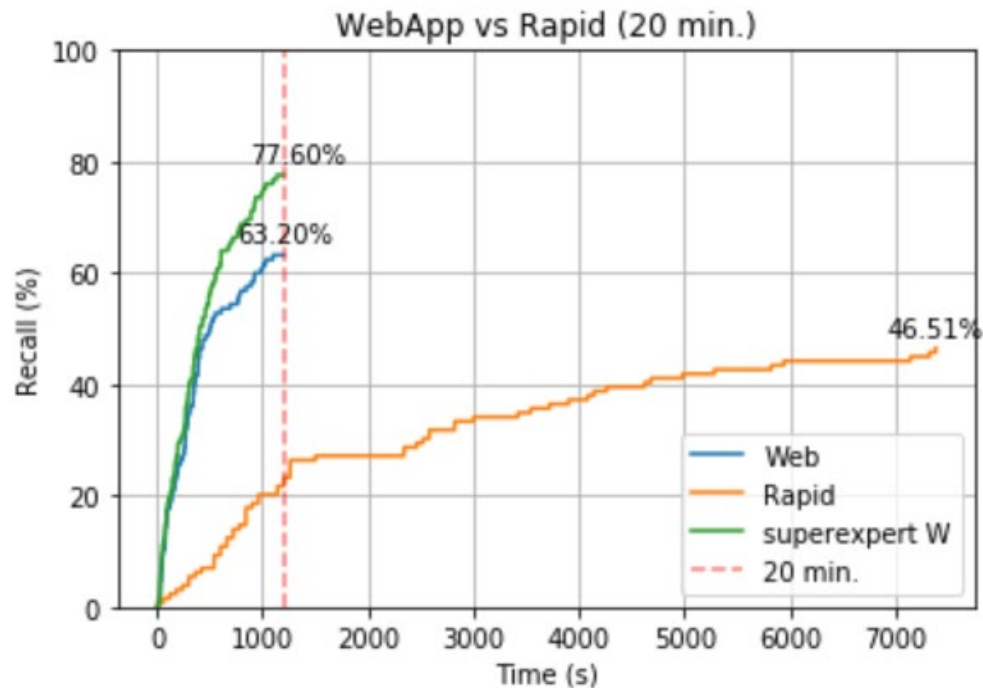
	000	002	007	023	031	039	040	062	066	067	084	085	088	096	101	114	024	052	077	090	095	117
B1(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B1(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
chiui1(R)	-	0/2	-	5/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	2/4	1/2	-	0/1	-	0/1	-
chiui1(W)	-	0/2	-	4/6	-	2/2	-	4/6	-	3/5	-	2/2	-	1/3	-	3/4	-	3/3	-	5/5	-	1/2
chiui2(R)	0/1	-	5/13	-	7/16	-	4/6	-	0/5	-	2/3	-	3/5	-	1/2	-	-	1/3	-	4/5	-	1/2
chiui2(W)	1/1	-	9/13	-	9/16	-	5/6	-	4/5	-	2/3	-	4/5	-	1/2	-	2/2	-	1/1	-	1/1	-
chiui3(R)	-	0/2	-	4/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	1/4	0/2	-	1/1	-	0/1	-
chiui3(W)	-	0/2	-	4/6	-	1/2	-	5/6	-	3/5	-	1/2	-	0/3	-	3/4	-	-	-	-	-	-

Results

	000	002	007	023	031	039	040	062	066	067	084	085	088	096	101	114	024	052	077	090	095	117
B1(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B1(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(R)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3(W)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
chiui1(R)	-	0/2	-	5/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	2/4	1/2	-	0/1	-	0/1	-
chiui1(W)	-	0/2	-	4/6	-	2/2	-	4/6	-	3/5	-	2/2	-	1/3	-	3/4	-	3/3	-	5/5	-	1/2
chiui2(R)	0/1	-	5/13	-	7/16	-	4/6	-	0/5	-	2/3	-	3/5	-	1/2	-	-	1/3	-	4/5	-	1/2
chiui2(W)	1/1	-	9/13	-	9/16	-	5/6	-	4/5	-	2/3	-	4/5	-	1/2	-	2/2	-	1/1	-	1/1	-
chiui3(R)	-	0/2	-	4/6	-	1/2	-	2/6	-	3/5	-	1/2	-	2/3	-	1/4	0/2	-	1/1	-	0/1	-
chiui3(W)	-	0/2	-	4/6	-	1/2	-	5/6	-	3/5	-	1/2	-	0/3	-	3/4	-	-	-	-	-	-

Results

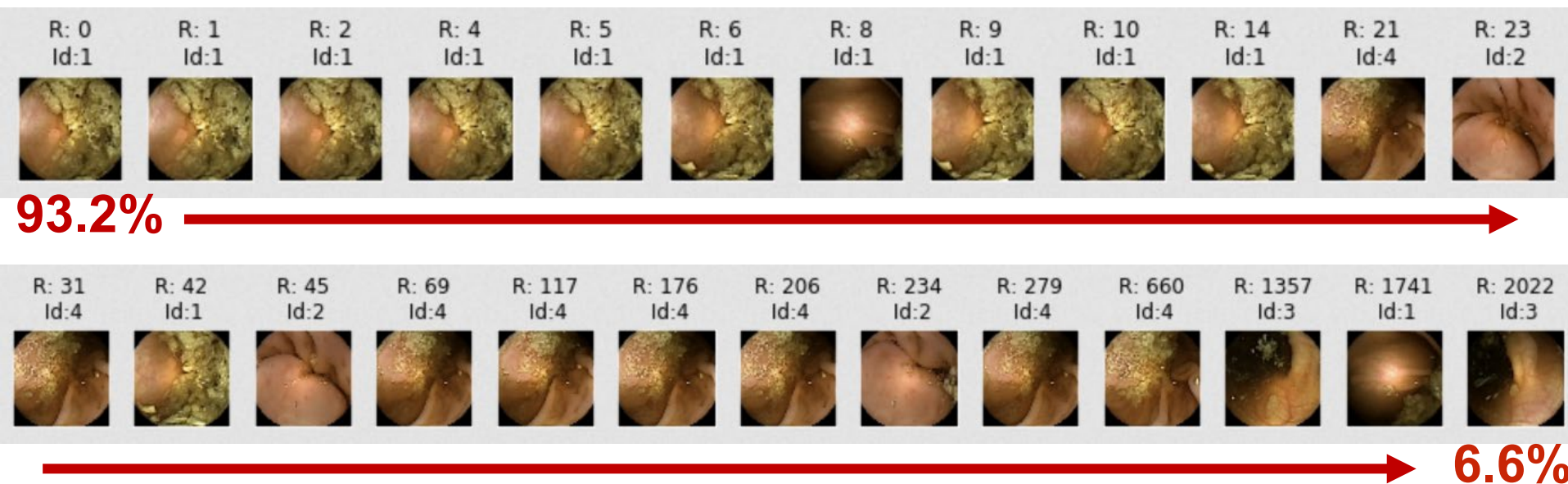
- **20 min** of app total use
- Time is reduced in a **factor of 6**
- **More polyps** detected when using the app



2. Learning to Sort

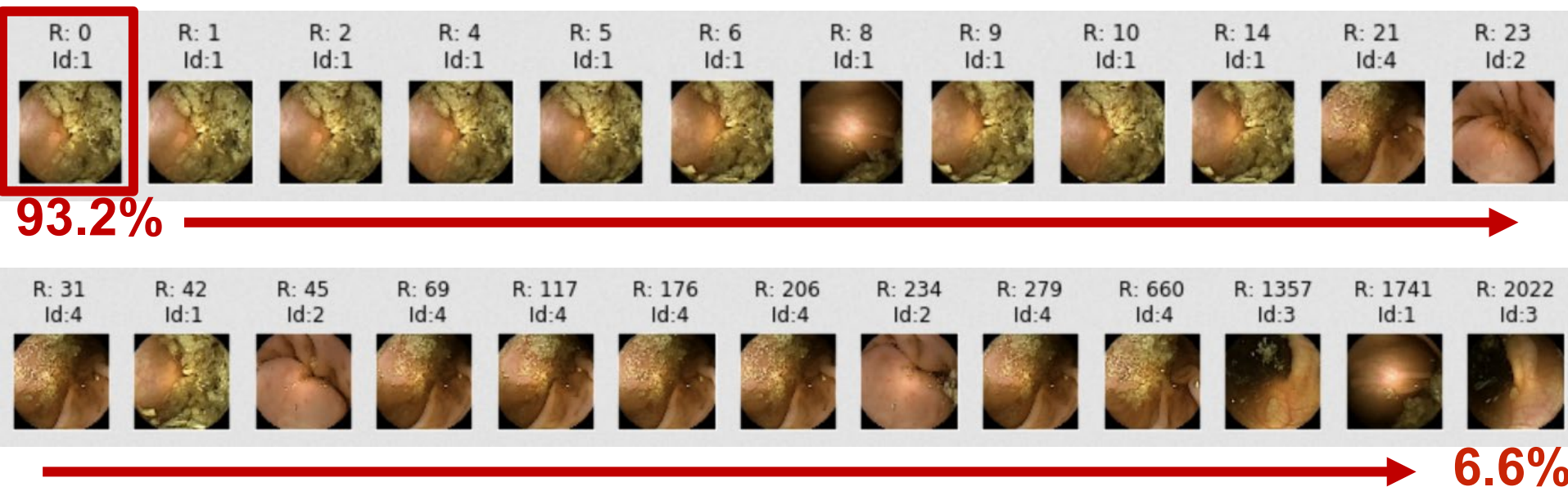
The Problem

Assuming that the network works as expected...



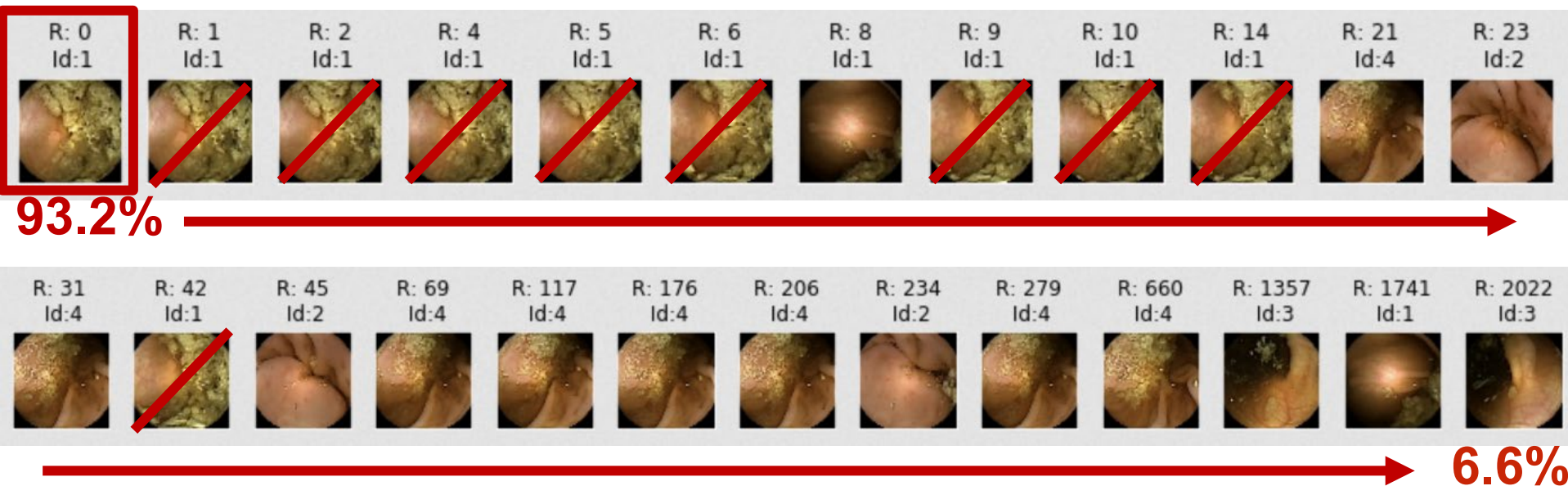
The Goal

Sort frames in a relevant way so the variability of polyps is show in early stages.



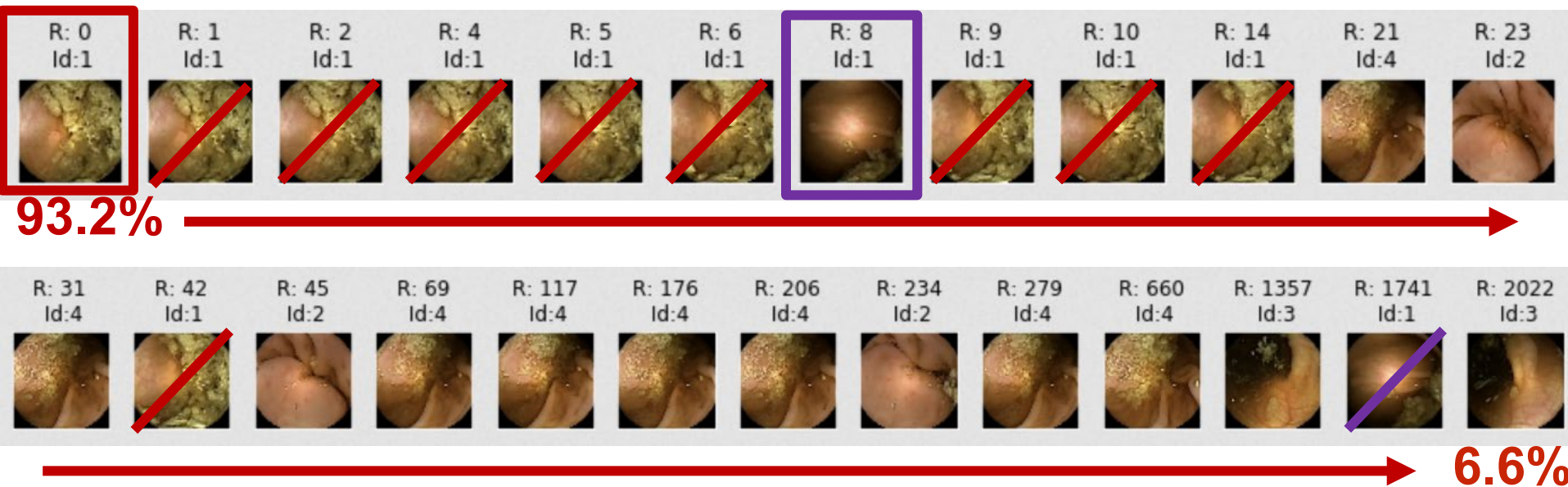
The Goal

Sort frames in a relevant way so the variability of polyps is show in early stages.



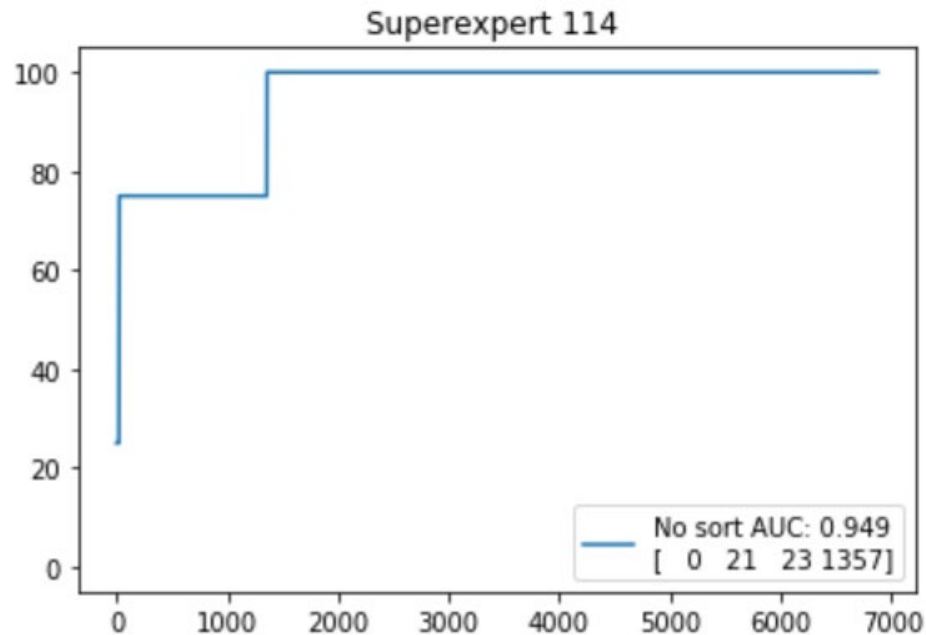
The Goal

Sort frames in a relevant way so the variability of polyps is show in early stages.



Superexpert

- Never misses a polyp
- Encoded in a superexpert curve



Objective function

Cost function can include a combination of these three metrics:

1. **Temporal** frame distance
2. **Embedding** distance
3. **Score** similarity

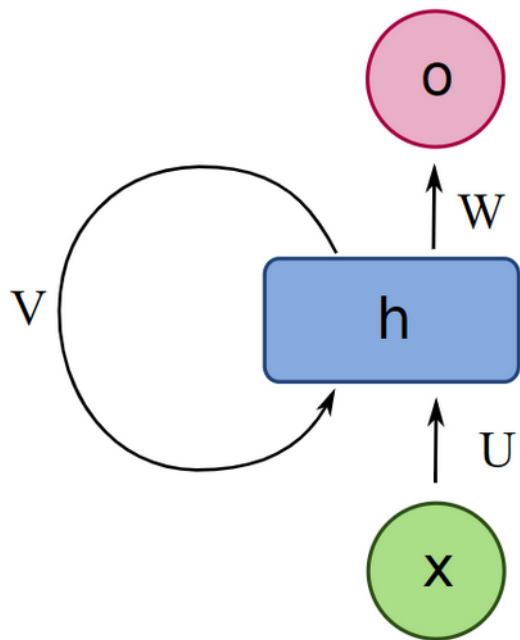
Some Ideas (1): Travel Salesman Problem (with memory)



Find the **minimum cost path** in the polyp graph

- Each node has a cost of:
$$\frac{1}{S_i}, i \in \{1, \dots, \#frames\}, S: score$$
- Each edge (i, j) has a cost of:
$$\frac{1}{d(emb_i, emb_j)}, \quad d: euclidean$$
- Restriction: Do not visit nodes similar to the ones seen in the past.

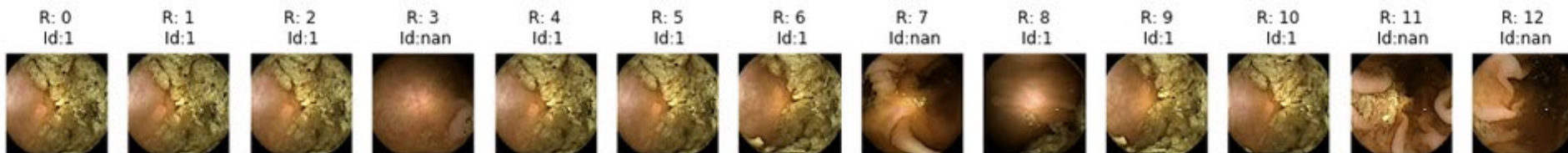
Some Ideas (2): Recurrent Neural Network



Learn to predict frames that are **different** from those seen so far but have a **high probability** of containing a polyp

Some Ideas (3): ResNet Embeddings (Greedy approach)

- Imagine that we have the following list of images that the physician is going to revise.



- This can be seen as a **priority queue** initially sorted by the probability of containing a polyp.

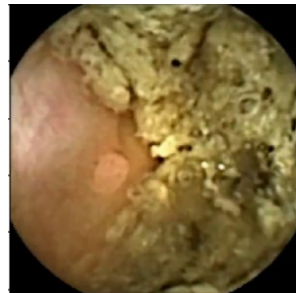
Some Ideas (3): ResNet Embeddings (Greedy approach)

1. Get the frame with the highest score



Some Ideas (3): ResNet Embeddings (Greedy approach)

1. Get the frame with the highest score
2. Compute its embedding and retrieve most similar images



R: 0
Id:1

R: 2
Id:1

R: 1
Id:1

R: 4
Id:1

R: 5
Id:1

R: 10
Id:1

R: 6
Id:1

R: 14
Id:1

R: 42
Id:1

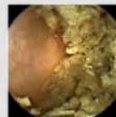
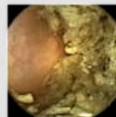
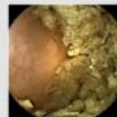
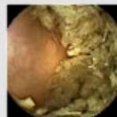
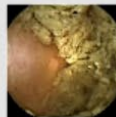
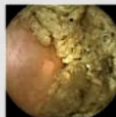
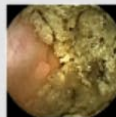
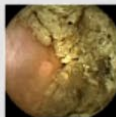
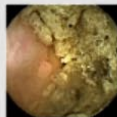
R: 9
Id:1

R: 6718
Id:nan

R: 6774
Id:nan

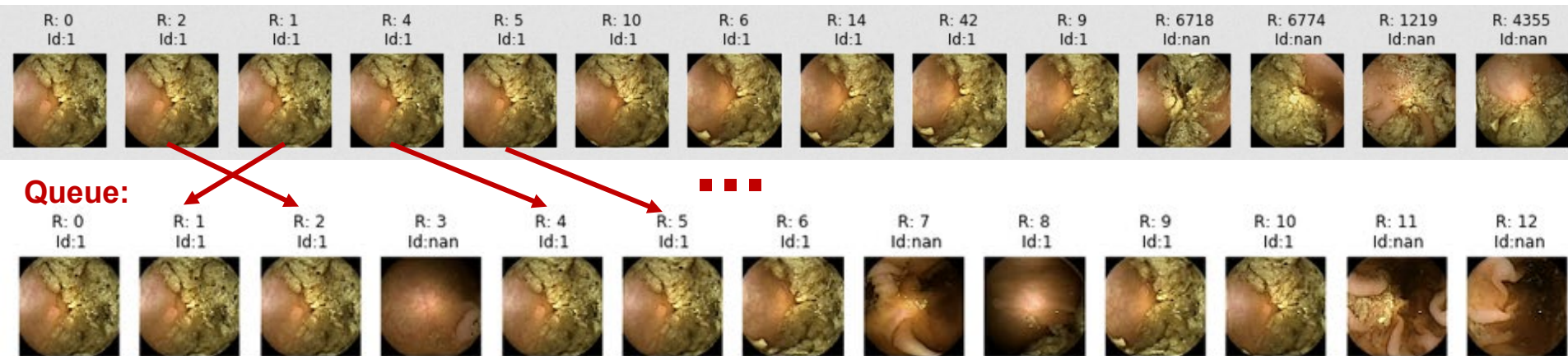
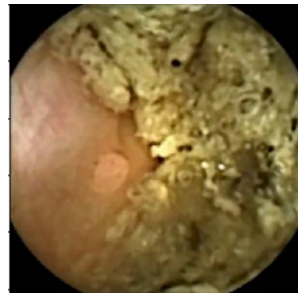
R: 1219
Id:nan

R: 4355
Id:nan



Some Ideas (3): ResNet Embeddings (Greedy approach)

1. Get the frame with the highest score
2. Compute its embedding and retrieve most similar images
3. Move images at the end of the queue

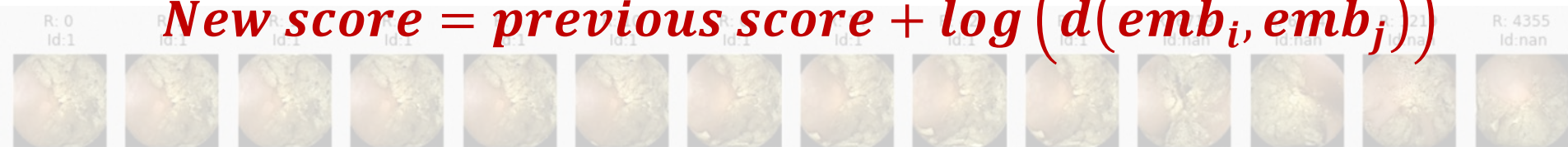


Some Ideas (3): ResNet Embeddings (Greedy approach)

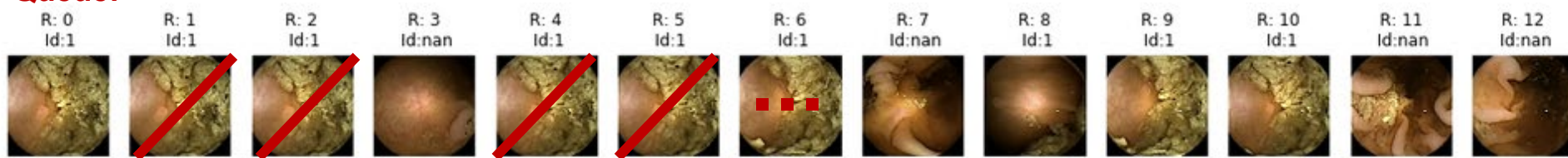
1. Get the frame with the highest score
2. Compute its embedding and retrieve most similar images
3. Move images at the end of the queue



$$\text{New score} = \text{previous score} + \log(d(\text{emb}_i, \text{emb}_j))$$



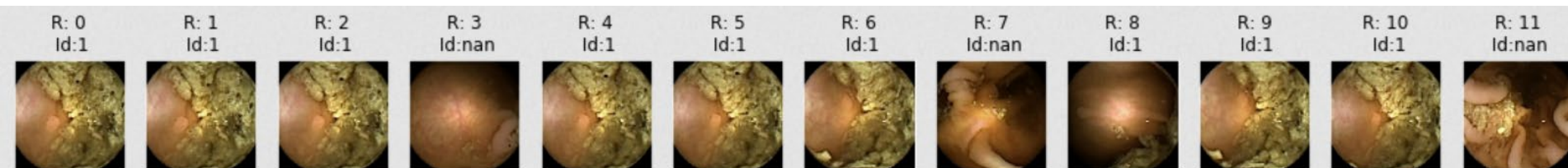
Queue:



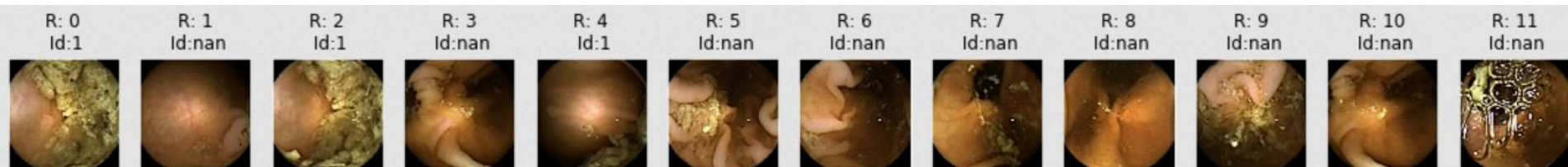
Some Ideas (3): ResNet Embeddings (Greedy approach)

Results

- Before sorting

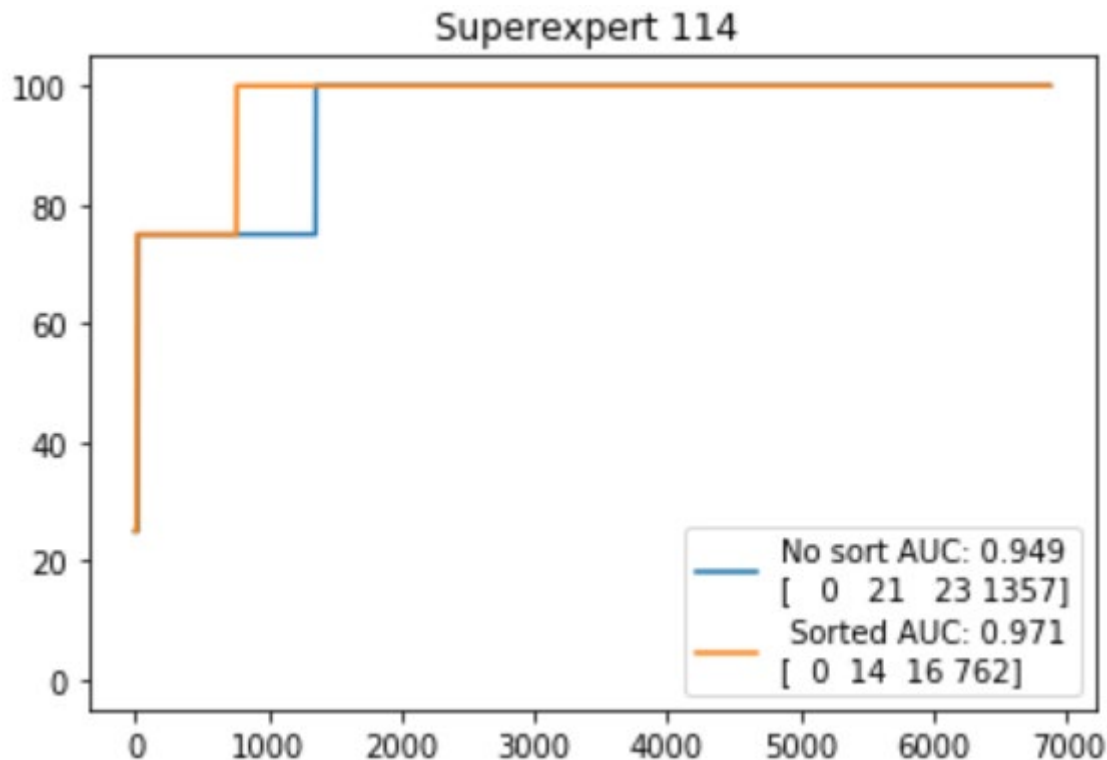


- After sorting



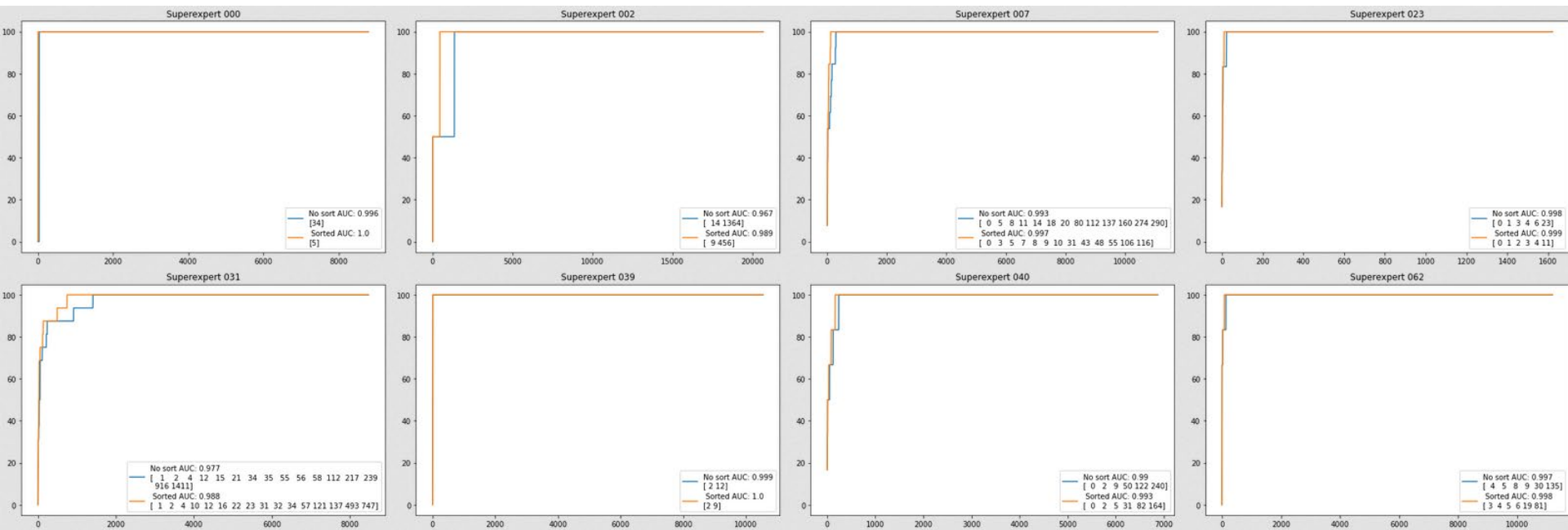
Some Ideas (3): ResNet Embeddings (Greedy approach)

Results



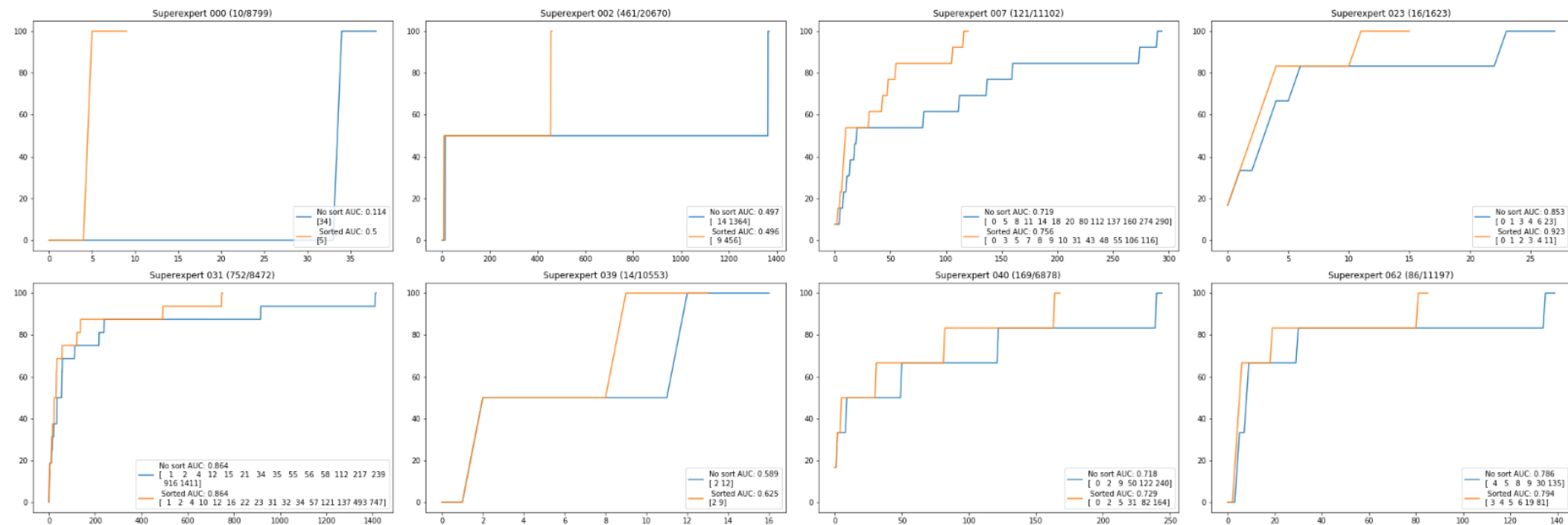
Some Ideas (3): ResNet Embeddings (Greedy approach)

Results



Some Ideas (3): ResNet Embeddings (Greedy approach)

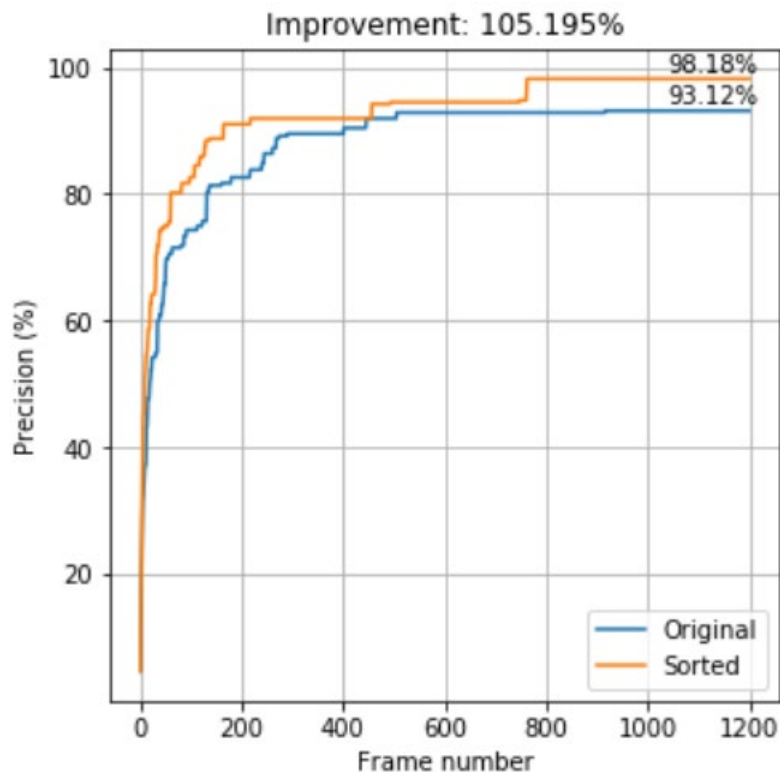
Results



Some Ideas (3): ResNet Embeddings (Greedy approach)

Results (mean)

$$100 \times \frac{AUC_{Sorted}}{AUC_{Original}} = 105.195$$



Thanks!

