VTT Technical Research Centre of Finland

Object detection for the analysis of creep voids in high-temperature metallic structures

Zeb, Akhtar; Tahkola, Mikko; Pohja, Rami; Pakarinen, Janne

Published: 24/11/2021

AI Day 2021

Object detection for the analysis of creep voids in high-temperature metallic structures

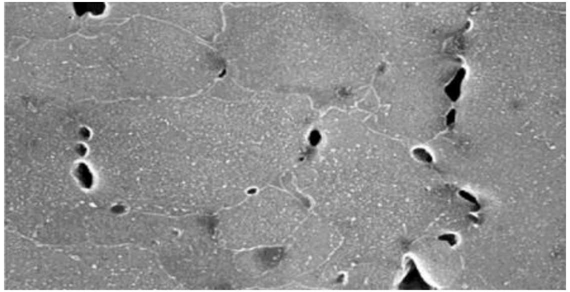
Akhtar Zeb, Mikko Tahkola, Rami Pohja, Janne Pakarinen

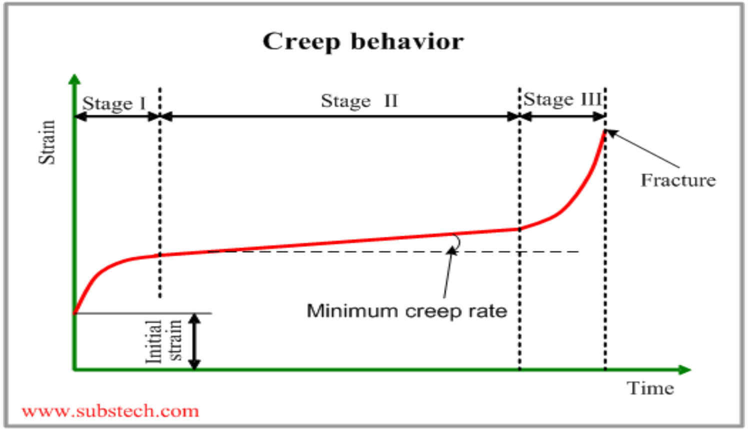
Problem:

Creep voids in metallic structures

Problem of creep void in metallic structures

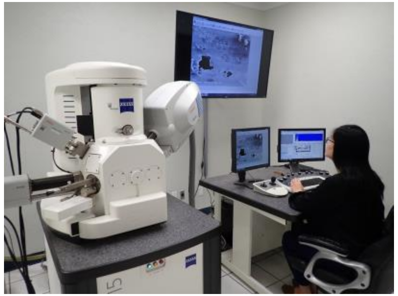
- High-temperature metallic structures under load experience creep at temperature about 0.3 times the material’s melting temperature
- Creep means significant viscous time-dependent and liquid-like material flows in the direction of principal stress
- Prolonged creep condition leads to rupture and component failure
- At relatively early stages of creep, one important manifestation is the creep void at the grain boundaries of the material
- Reliable and accurate detection of creep void size and density helps in improving the life cycle management of high-temperature components





Creep void analysis – traditional vs machine learning approach

- Creep void inspection in service conditions is usually performed by replica inspection
- Replica inspection samples are taken from components and inspected via optical and/or scanning electron microscope (SEM)
- However, interpretation of the results is often difficult and time-consuming
- ML can help in reducing human errors and speed up the process by analyzing large areas and multiple sample images



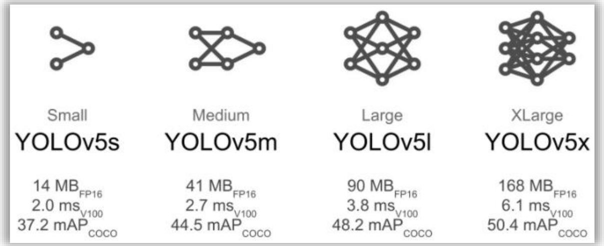
Our solution:

YOLO object detection algorithm

YOLO algorithm

- **Y**ou **O**nly **L**ook **O**nce
- Object detection algorithm popular for its speed, and outperformed, e.g., *Sliding Window Object Detection*, *R CNN*, *Fast R CNN*, *Faster R CNN* algorithms
- Built on PyTorch framework and tells about object class and object localization, i.e. bounding box
- Makes all the predictions in one forward pass
- Yolo v1, Yolo v2, Yolo v3, Yolo v4, Yolo v5
- **YOLOv5** is a family of object detection architectures and models pretrained on the COCO dataset, and represents *Ultralytics*[1] open-source research into future vision AI methods
- We use the YOLOv5s, which is the smallest and fastest model
- We take advantage of pre-trained weights and default configuration of YOLOV5s

[1]https://ultralytics.com/



Pretrained Checkpoints

Model	size (pixels)	mAP^{val} 0.5:0.95	mAP^{test} 0.5:0.95	mAP^{val} 0.5	Speed V100 (ms)		params (M)	FLOPs 640 (B)
YOLOv5s	640	36.7	36.7	55.4	**2.0**		7.3	17.0
YOLOv5m	640	44.5	44.5	63.1	2.7		21.4	51.3
YOLOv5l	640	48.2	48.2	66.9	3.8		47.0	115.4
YOLOv5x	640	**50.4**	**50.4**	**68.8**	6.1		87.7	218.8
YOLOv5s6	1280	43.3	43.3	61.9	**4.3**		12.7	17.4
YOLOv5m6	1280	50.5	50.5	68.7	8.4		35.9	52.4
YOLOv5l6	1280	53.4	53.4	71.1	12.3		77.2	117.7
YOLOv5x6	1280	**54.4**	**54.4**	**72.0**	22.4		141.8	222.9
YOLOv5x6 TTA	1280	**55.0**	**55.0**	**72.0**	70.8		-	-

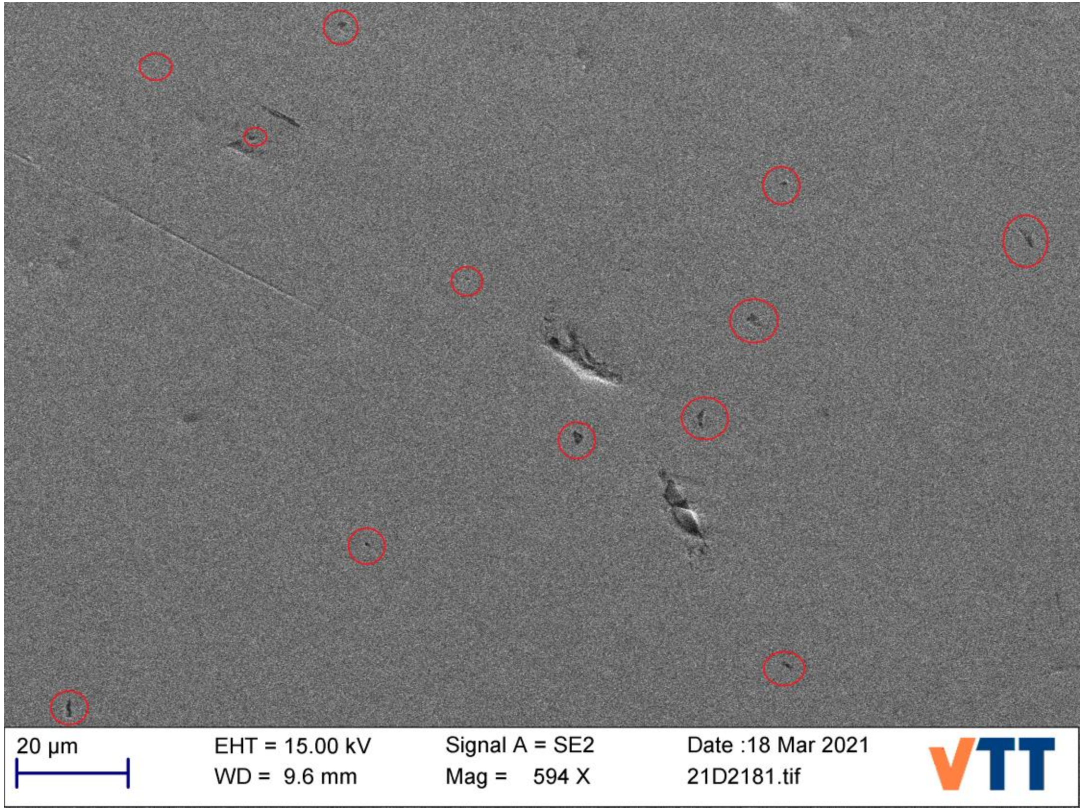
Source: https://github.com/ultralytics/yolov5

Data:

Scanning electron microscope images

Scanning electron microscope images

- 100 SEM images of oxygen-free phosphorous doped copper sample surfaces
- With creep voids of different shapes and sizes
- Distinguishing creep voids from other damages could be challenging task even for the material expert

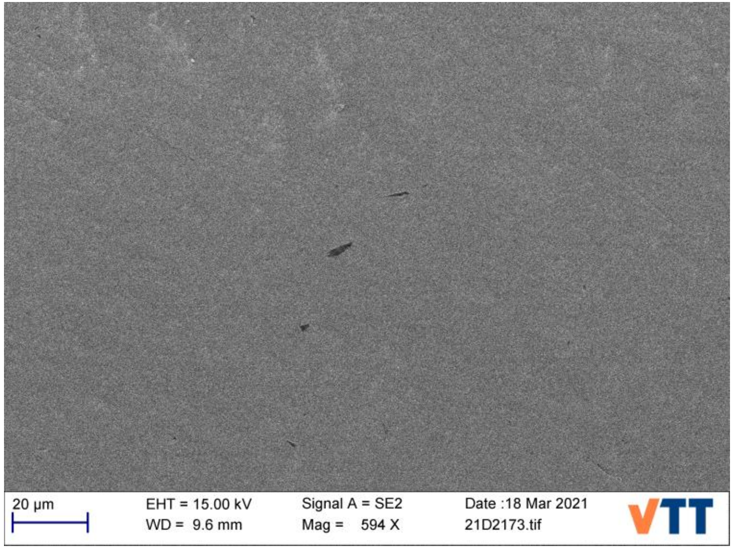


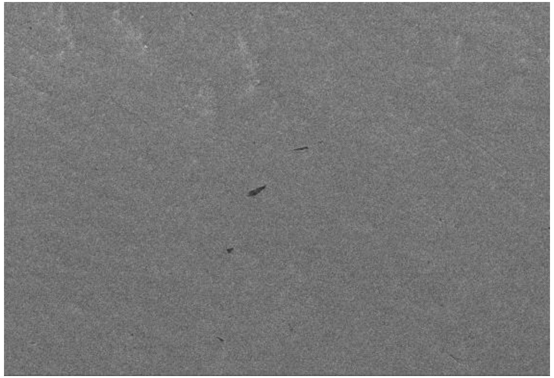
Data pre-processing and annotation

Pre-processing images

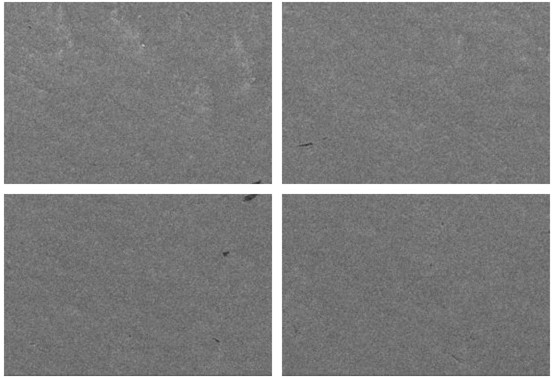
- Formatting and cropping
- Resizing
- Splitting

(height, width) (2304, 3072)





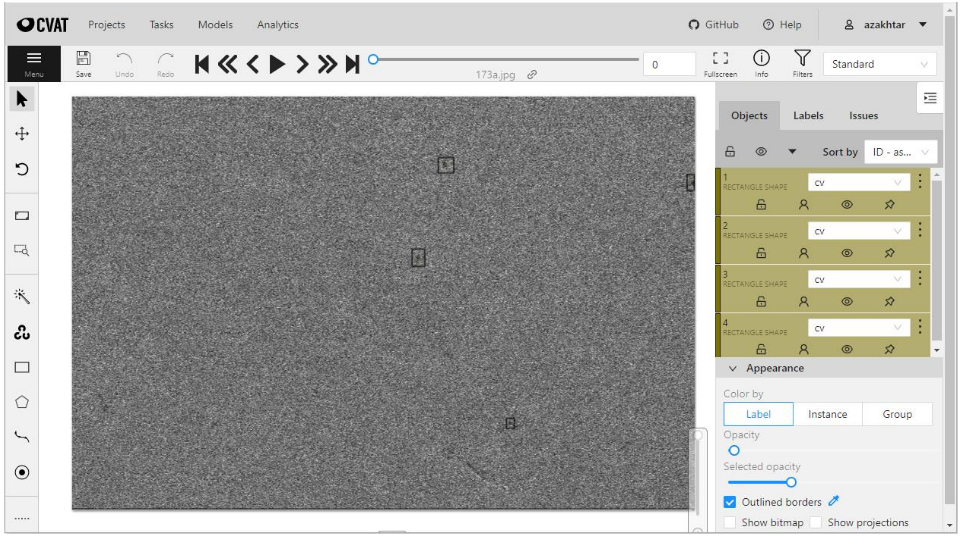
(height, width)
(2048, 3072)

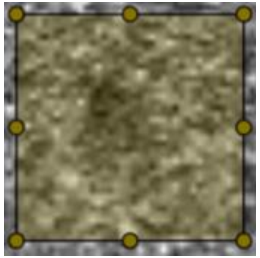


4x
(height, width)
(1024, 1536)

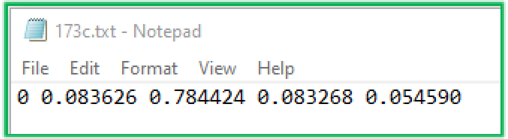
Labelling and annotation

- CVAT – computer vision annotation tool
- Free, online, interactive video and image annotation tool for computer vision
- Supports several annotation formats (coco, yolo, etc.)
- Only one class “cv”, in this study





Yolo format:
Bounding box coordinates are in normalized xywh format (from 0 -1)
Each row is **class** **x_center** **y_center** **width** **height** format



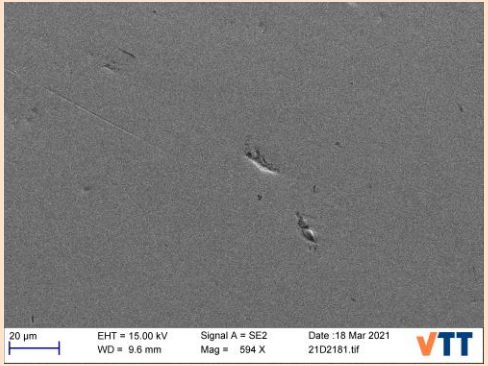
YOLO training, inference, and calculations

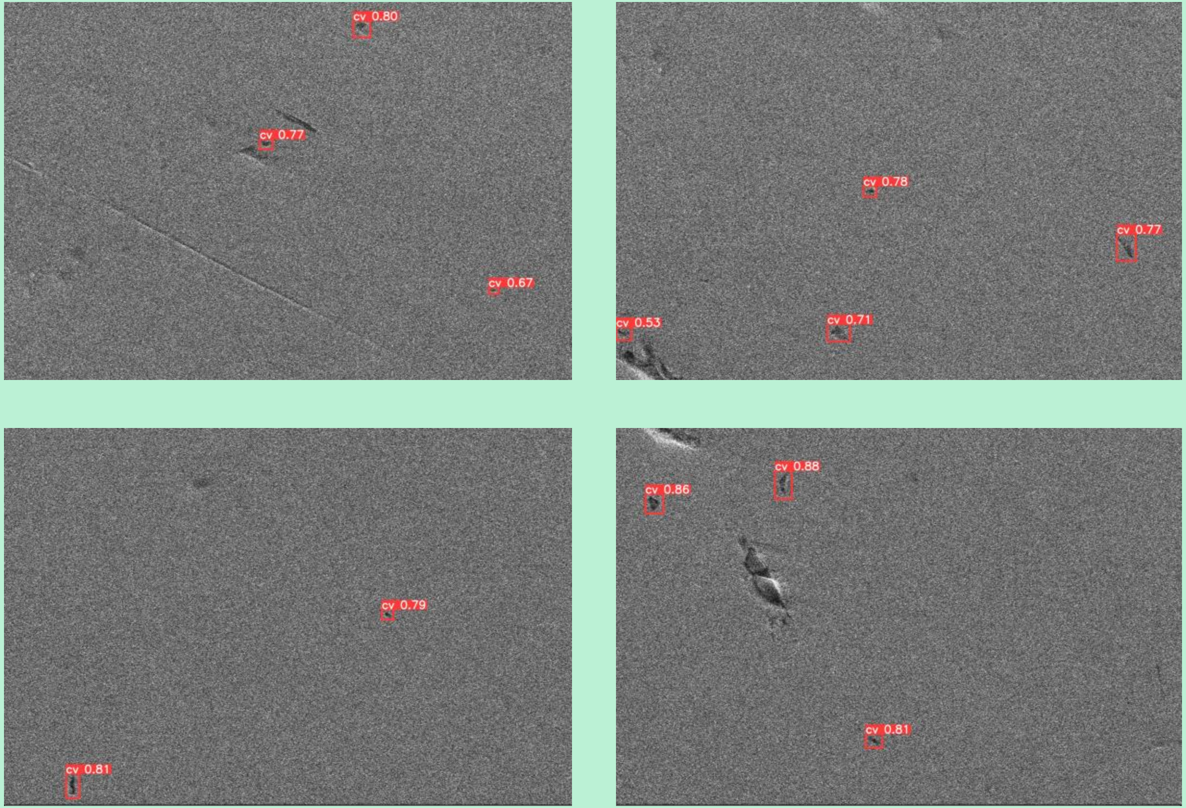
Training and results

- Google Colab
- Images annotated using CVAT and exported in yolo format
- YOLOv5s model is trained using pretrained weights and default configuration and architecture

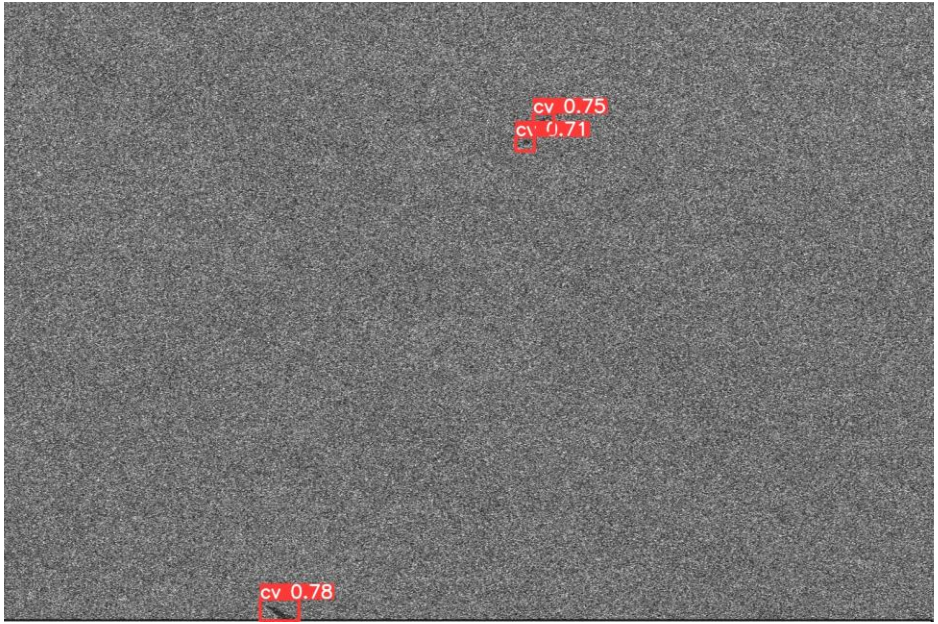
Images	Epochs	Training time	mAP_0.5
64	100	45 min	0.85
64	150	1 hour	0.83
200	100	2 hours	0.82

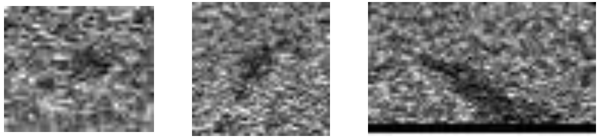
YOLOv5 inference – detections on full-size images



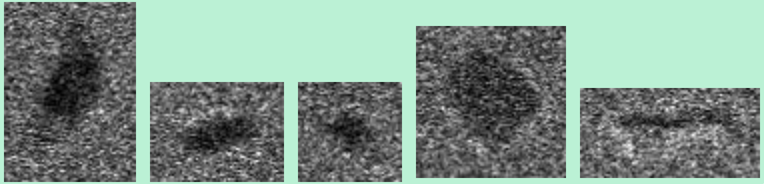


YOLOv5 inference – bounding boxes and creep void images



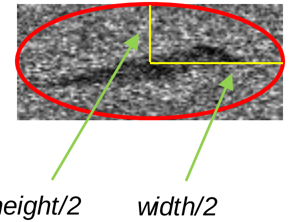


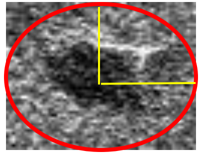
[	xcenter	ycenter	width	height	confidence	class	name
0	455.384216	1004.244934	63.769897	35.510864	0.779910	0	cv
1	890.653381	200.352341	34.046265	33.055939	0.751255	0	cv
2	860.605896	234.063416	30.840942	24.863892	0.712342	0	cv



Creep void calculations

- cv_area = $\pi \times width/2 \times height/2$
- cv_num = *len(cv_area)*
- cv_frac_area = *sum(cv_area)/image(w*h)*100*





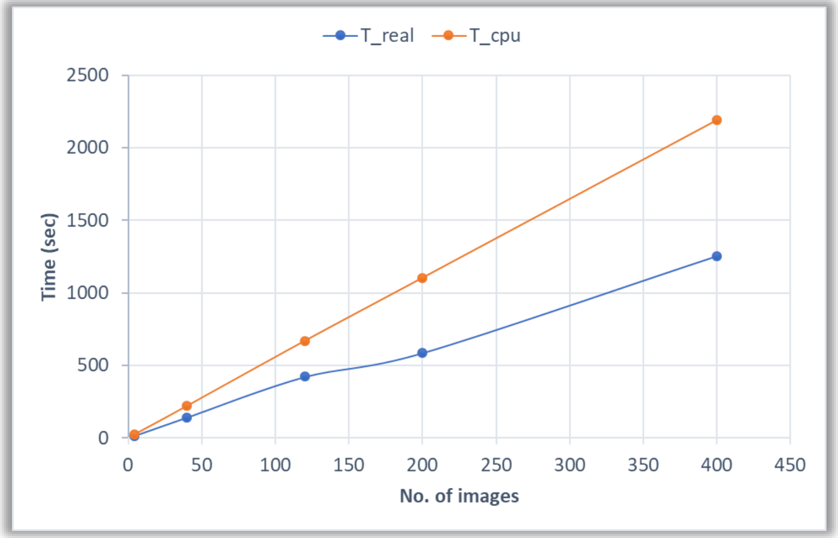
```
Number of creep-voids in an image: [6, 1, 6, 5, 3, 0, 0, 3]

Fractional area (percentage) of (all) creep-voids in an image: [0.3635374031375304,
0.10300326269379431, 0.4792252592922566, 0.21302700214886802, 0.23649661238944594, 0.0, 0.0, 0.14832590684477895]

Area of creep-void in an image:
 [0     788.529447
1     599.364558
2    1353.928740
3     559.977414
4     550.245897
5    1865.902885
Name: height, dtype: float64, 0    1620.101238
Name: height, dtype: float64, 0    1795.564032
1    1478.330111
2     669.510445
3    2322.489817
4     435.992353
5     835.674824
Name: height, dtype: float64, 0    933.012768
1    960.984515
2    344.420429
3    606.764116
4    505.443199
Name: height, dtype: float64, 0    3111.030319
1     385.109797
2     223.629962
Name: height, dtype: float64, Series([], Name: height, dtype: object), Series([], Name: height, dtype: object), 0    989.901862
1    857.765641
2    485.297288
Name: height, dtype: float64]
```

Inference time

- Total time taken by the algorithm starting from the initial pre-processing of the images to the saving of final results (Intel® Core™ i7-8665U Processor, 16 GB RAM)
- T_real is the system time in seconds from the starting point to the end of a process
- T_cpu is the sum of the system and user CPU time from the beginning to the end of a process



Conclusions

Conclusions and future work

- YOLOv5 built on PyTorch framework
- Easy to train, e.g. using free Google Colab environment including GPUs
- Local installation is straightforward
- Cropped images of the creep voids are obtained, in addition to the detections on full-size images
- Detected all the creep voids correctly
- Bounding boxes coordinates enable calculation of desired parameters
- Labelling accuracy would result in better inferences
- Creep void images can be further investigated using other tools, e.g. ImageJ

beyond

the obvious

First Name Surname
firstname.surname@vtt.fi

@VTTFinland
@your_account

www.vtt.fi