# A deep learning model for automated quantification of DNA repair foci in somatic mammalian cells

Sara Shadmehri[1], Tatevik Bezhanyan[1,2], Maksim Bondarev[2], Oksana Streltsova[1,2],
Maxim Zuev[1], Anna Chigasova[3], Andrey Osipov[3], Natalia Vorobyeva[3], Andreyan N. Osipov[3]

[1]Meshcheryakov Laboratory of Information Technologies, JINR , Dubna, Russia
[2]Dubna State University, Dubna, Russia
[3]Burnasyan Federal Medical Biophysical Center (SRC – FMBC), Moscow, Russia

**Yerevan, 21-10-2024**

This work was carried out within the state assignment of Ministry of Science and Higher Education of the Russian Federation (theme No. 124092700007-4).

# DNA Repair Foci

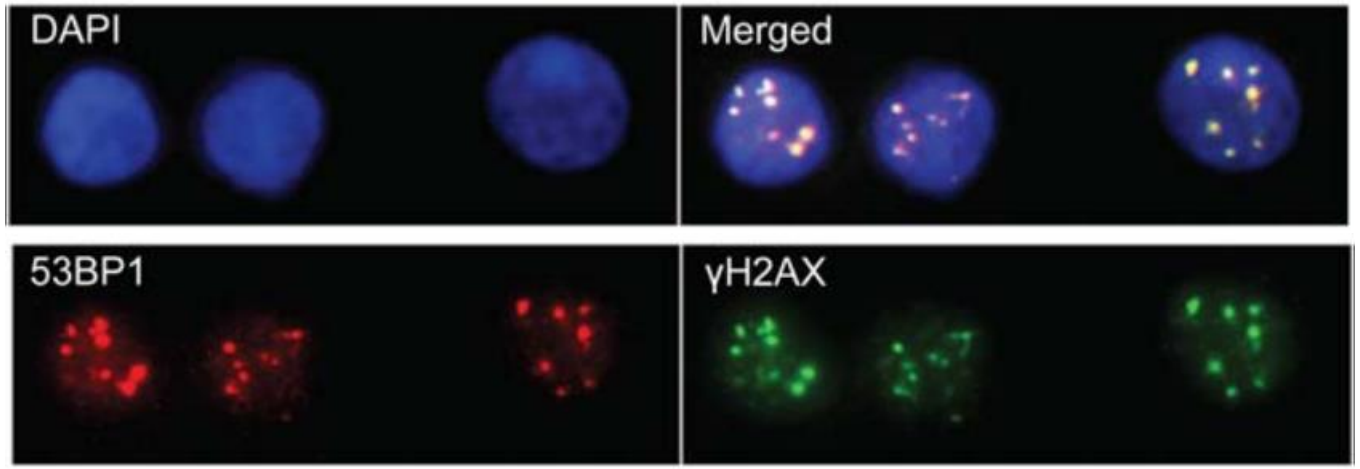
## Ionizing Radiation-Induced Foci (IRIF)

**DNA double-strand breaks (DSB):** The most deadly DNA damage induced by ionizing radiations

**DSB quantification :** a tool to analyze the damaging effect of radiation exposure

**Immunofluorescent staining method:** Revealing the specific protein biomarkers involved in the process of DNA DSB repair where the proteins accumulate at the sites of DNA DSBs to form foci

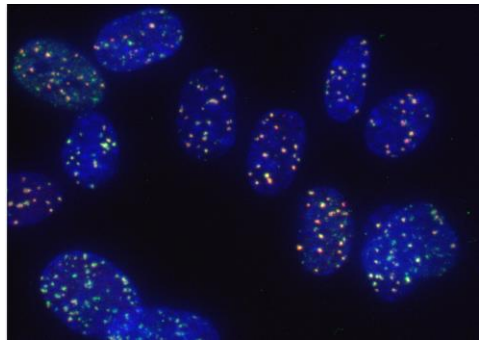
**γ-H2AX and 53BP1:** the most reliable biomarkers of DSBs




Human blood lymphocytes 24h after 4Gy X-irradiation ex vivo

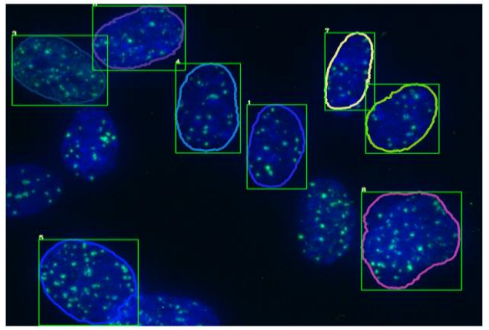
# Overview of the Foci Detector Algorithm

Upload the Image File → Marking the Cell Nuclei → Extracting Cell Images → Marking the Foci per Cell

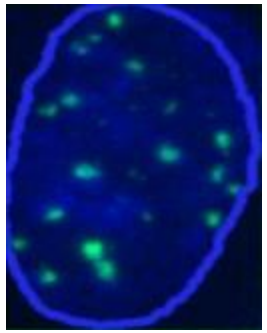


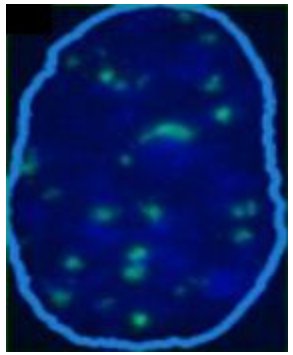
Cell Detection Neural Network Algorithm, OpenCV

Deleting Small and Boundary Cells

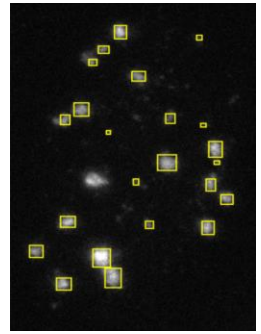


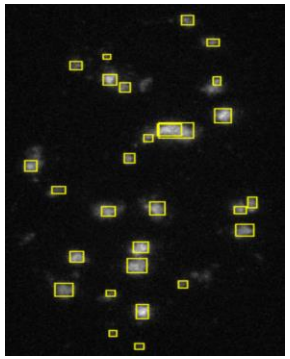
OpenCV





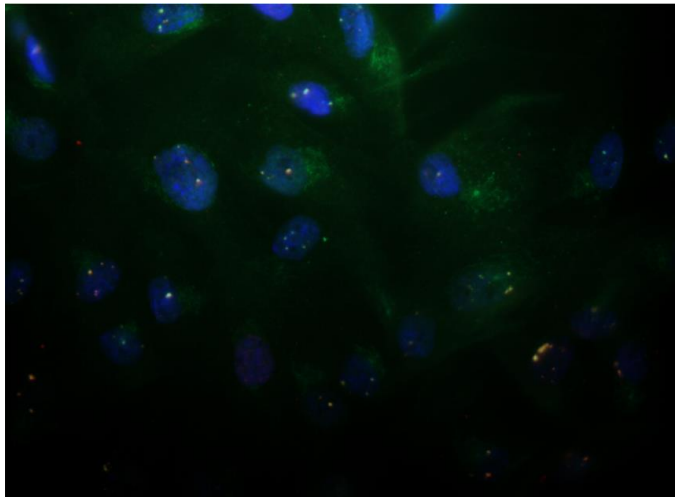
Neural Network based on YOLO Algorithm





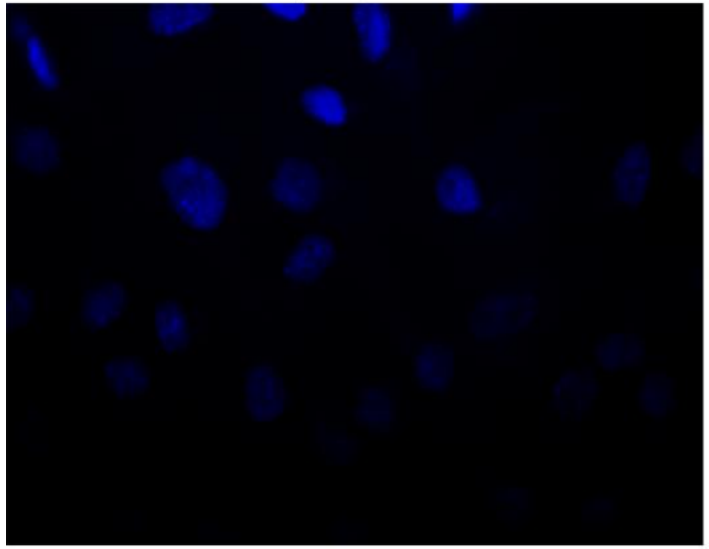
R. Wanotayan et al. , Scientific Reports **12**, 5527 ( 2022)

# First Stage: Cell Detection

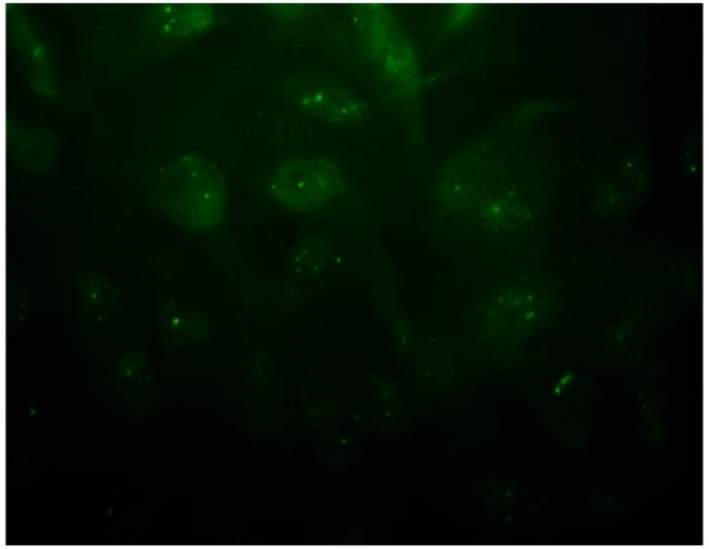


**Separating the color channels**

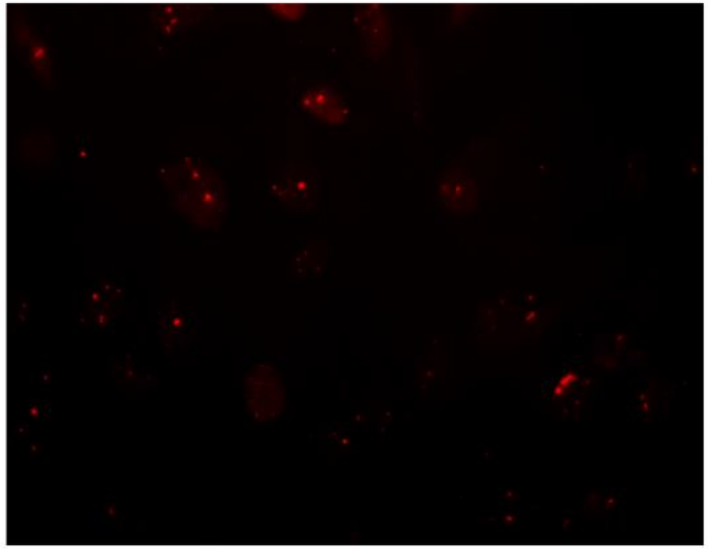
Blue (DAPI) channel



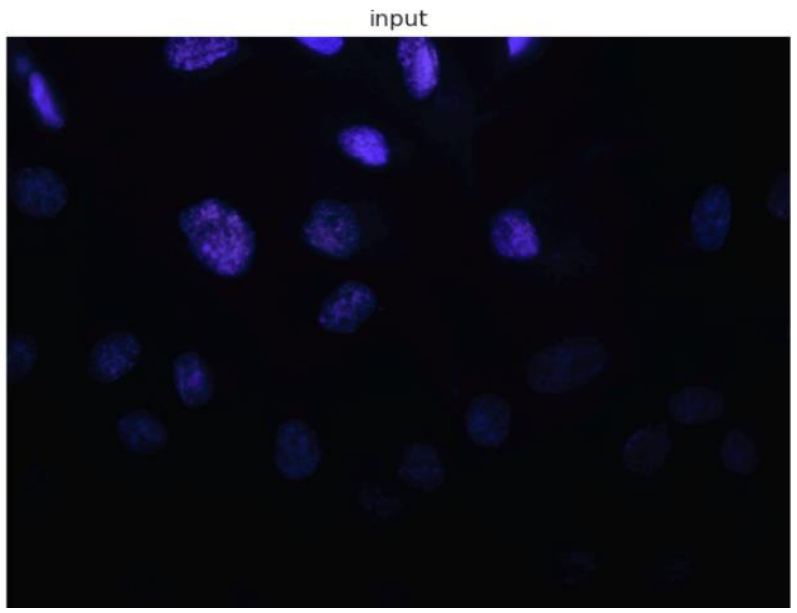
Green channel



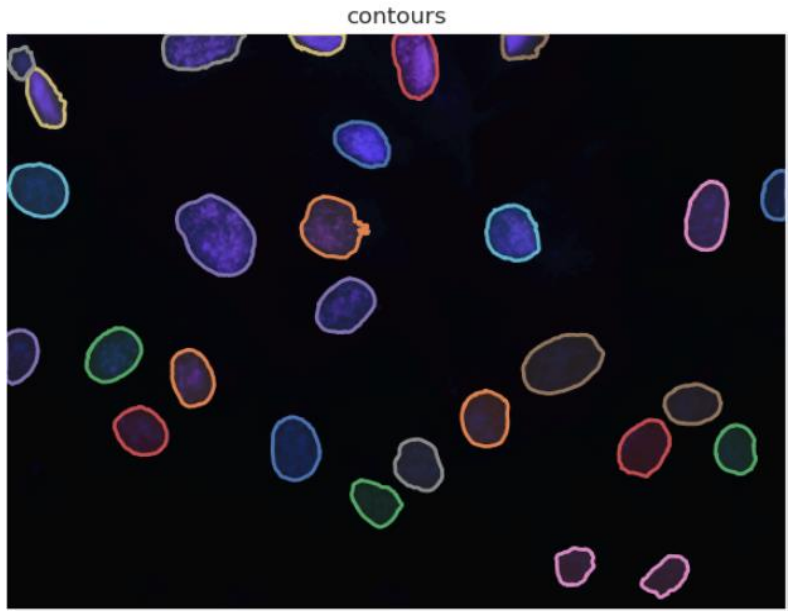
Red channel



# Find and Mark the Cells by Contours




Pretrained
Neural Network Model




https://github.com/FZJ-INM1-BDA/celldetection

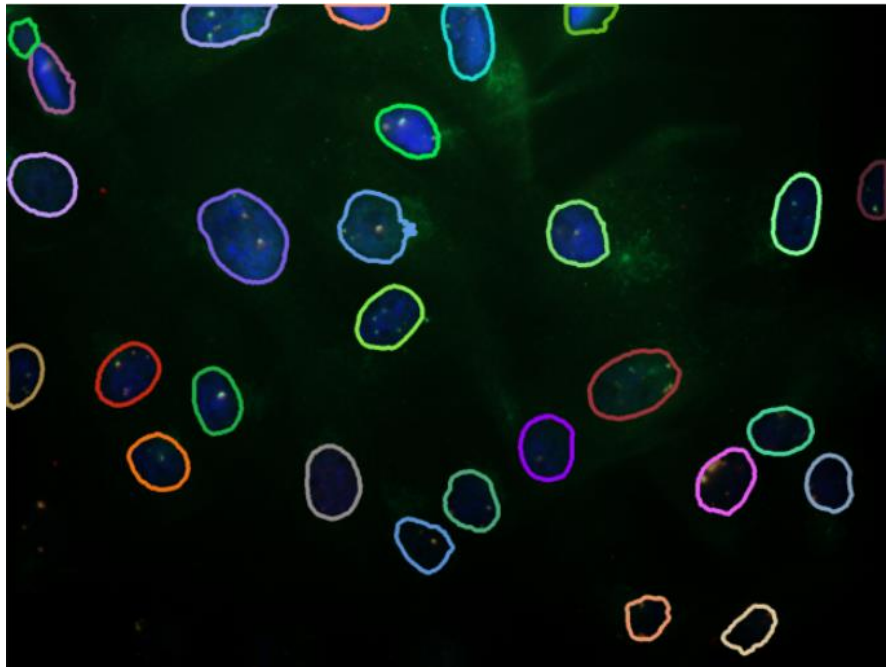
```python
import torch, cv2, celldetection as cd
```

```python
# Load pretrained model
device = 'cuda' if torch.cuda.is_available() else 'cpu'
model = cd.fetch_model('ginoro_CpnResNeXt101UNet-fbe875f1a3e5ce2c', check_hash=True).to(device);
model.eval();
```

```python
# Run model
with torch.no_grad():
    x = cd.to_tensor(img_b2, transpose=True, device=device, dtype=torch.float32)
    x = x / 255  # ensure 0..1 range
    x = x[None]  # add batch dimension: Tensor[3, h, w] -> Tensor[1, 3, h, w]
    y = model(x)
```

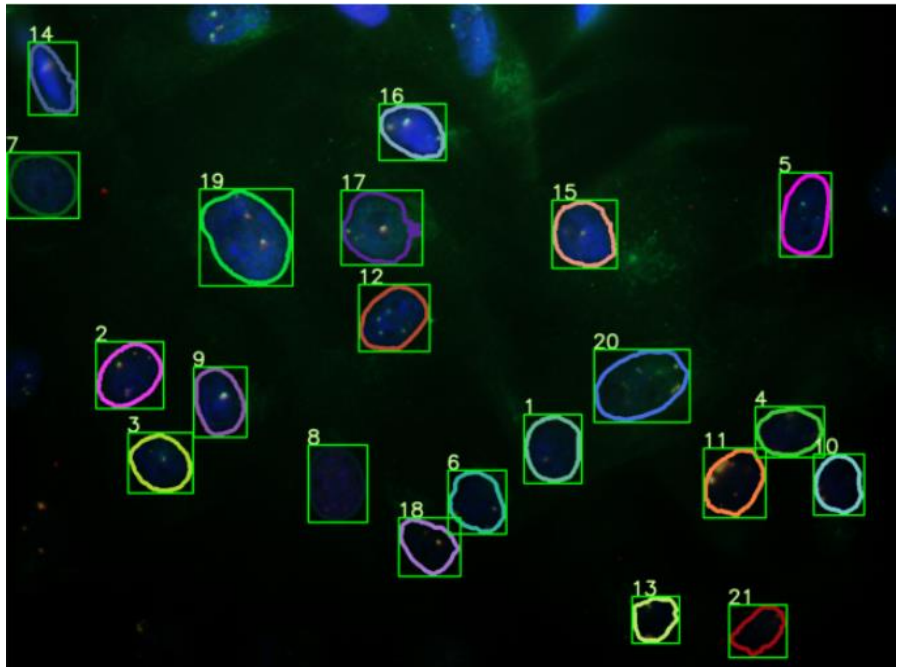
```python
# Show results for each batch item
contours = y['contours']
for n in range(len(x)):
    cd.imshow_row(x[n], x[n], figsize=(16, 9), titles=('input', 'contours'))
    cd.plot_contours(contours[n])
    plt.show()
```

E. Upschulte et al. , Medical Image Analysis **77**, 102371 (2022)

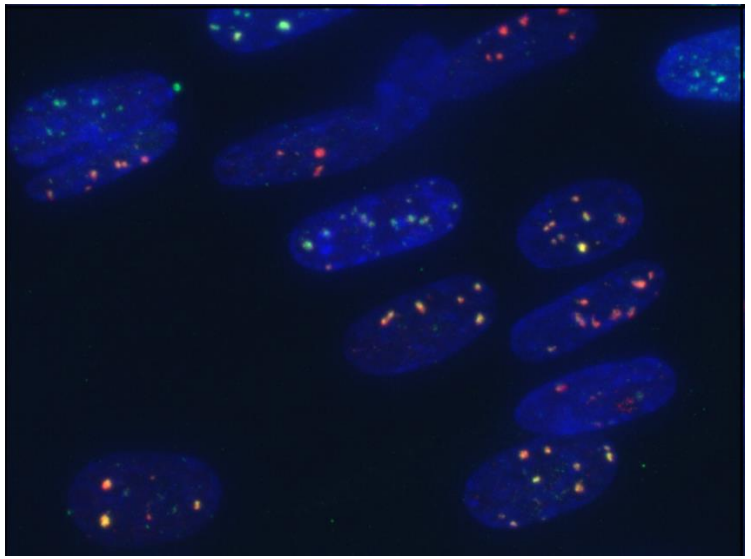


Creating DataFrame for
Bounding Box Coordinates

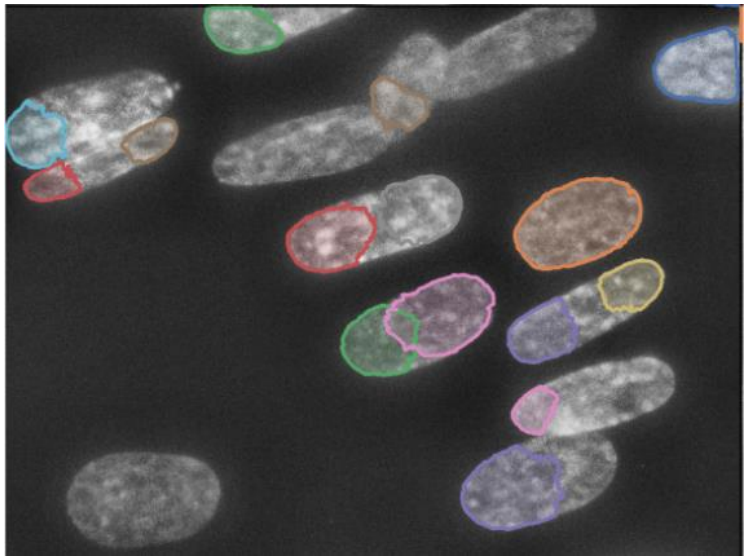
Deleting Small
and Boundary Objects



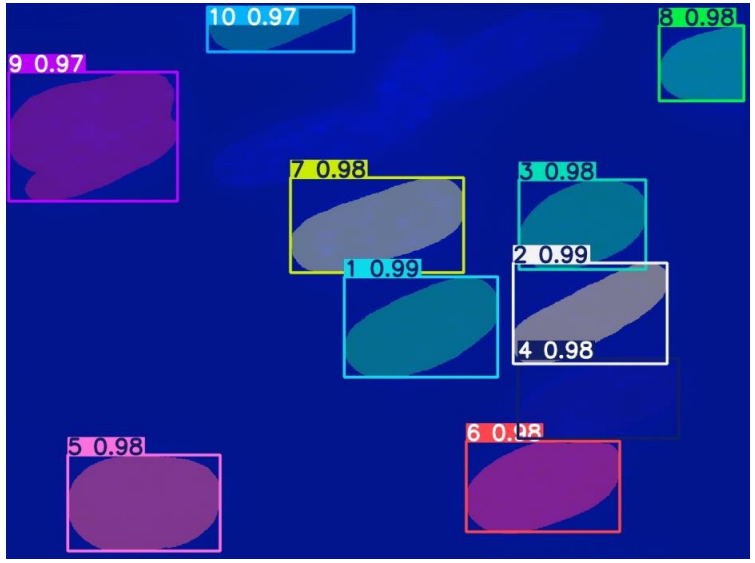
## Image



## "Celldetection" contours

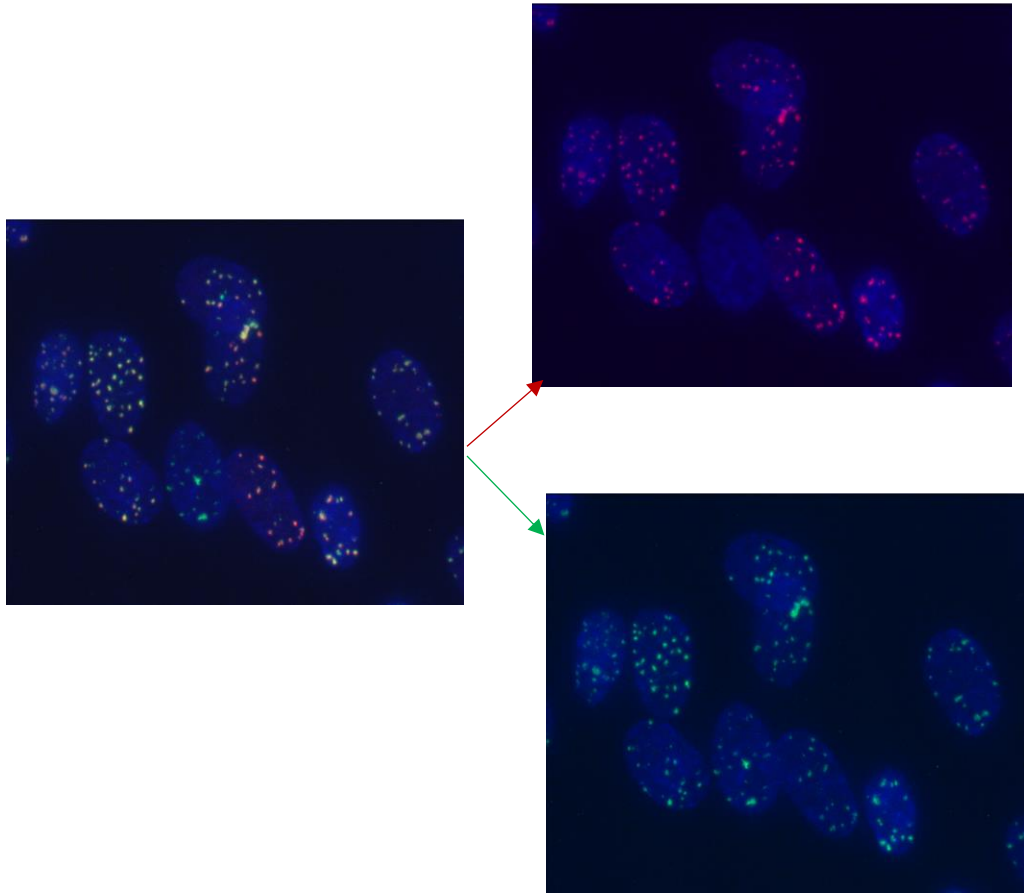


## SAM (Segment Anything Model) - Ultralytics

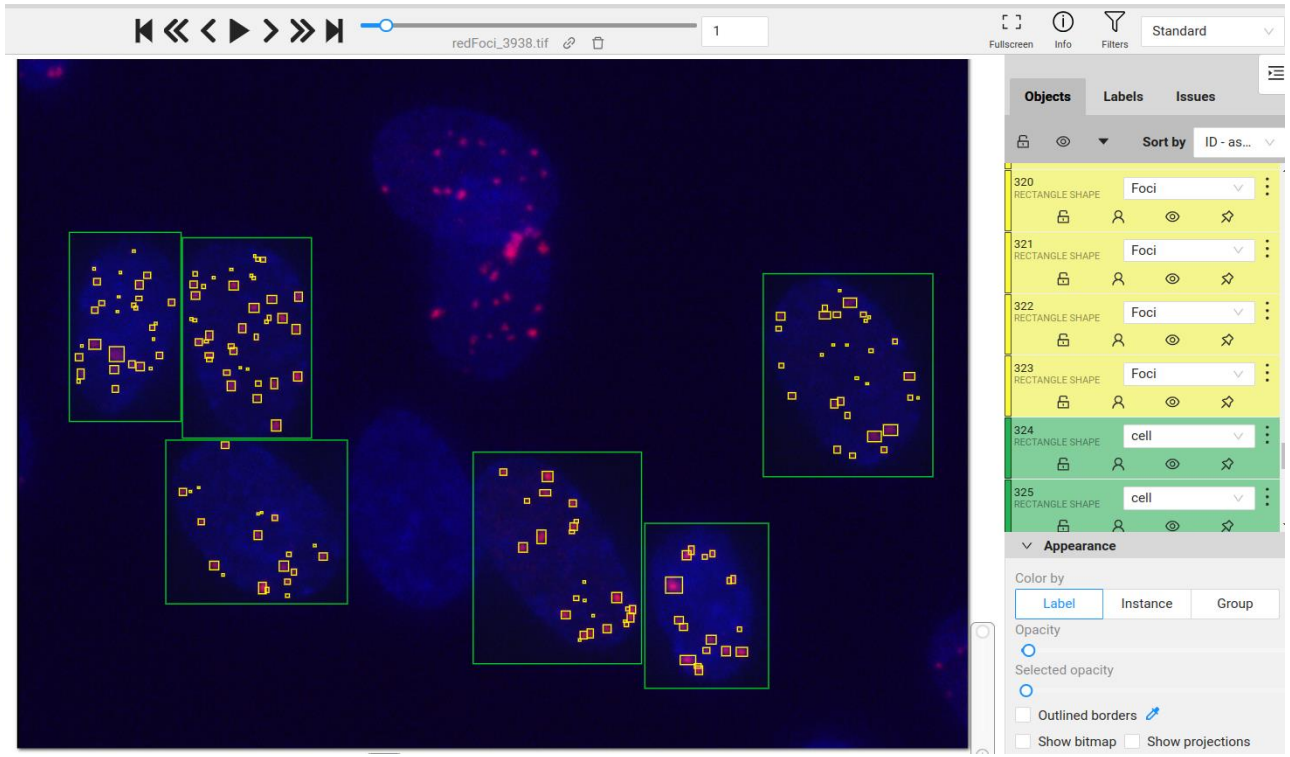


# Second Stage: Foci Detection in each cell

1) Annotating the Image Data as two classes; Foci and Cell



- Data Annotation on CVAT in Yolo format



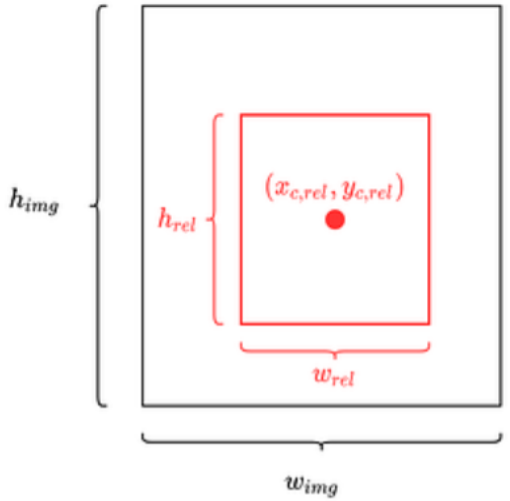

CVAT is installed on the ML/DL/HPC ecosystem of the HybriLIT Heterogeneous Computing Platform.

## 2) Cropping the annotated images to separate cells

### YOLO markup

```
X_CENTER_rel = X_CENTER_abs/IMAGE_WIDTH
Y_CENTER_rel = Y_CENTER_abs/IMAGE_HEIGHT
WIDTH_rel = WIDTH_OF_LABEL_abs/IMAGE_WIDTH
HEIGHT_rel = HEIGHT_OF_LABEL_abs/IMAGE_HEIGHT

 W = img.shape[1]
 H = img.shape[0]
```



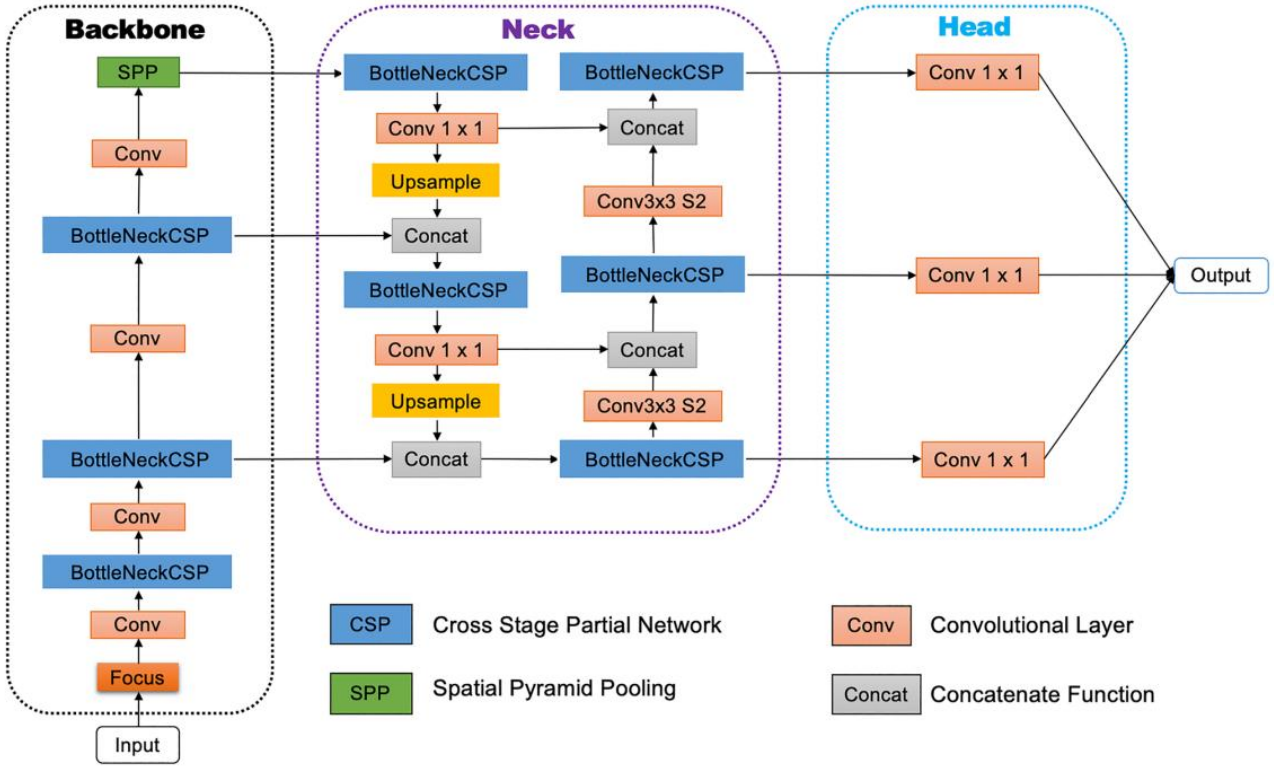

## 3) Reducing the classes to only one class: Foci

**Class renumbering to 0**

```
def file_write_label_img(dataset_src_img, vname, dataset_only_foci_img,
                         dataset_only_foci_labels, ind_cell, img_cell, X_label ):
    print(os.path.join(dataset_src_img, vname))
    base = os.path.splitext(vname)[0]
    print(base)

    img_name = os.path.join(dataset_only_foci_img, base +"_"+str(ind_cell)+".jpeg")
    print(img_name)
    cv2.imwrite(img_name, img_cell)
    label_name = os.path.join( dataset_only_foci_labels, base +"_"+str(ind_cell) +".txt")
    df_cell = pd.DataFrame(X_label)
    df_cell[0] = 0   ############ class == 0
    print(df_cell)
    df_cell.to_csv(label_name , index=False,  header=None, sep='\t')
```

# Yolov5 Network Architecture

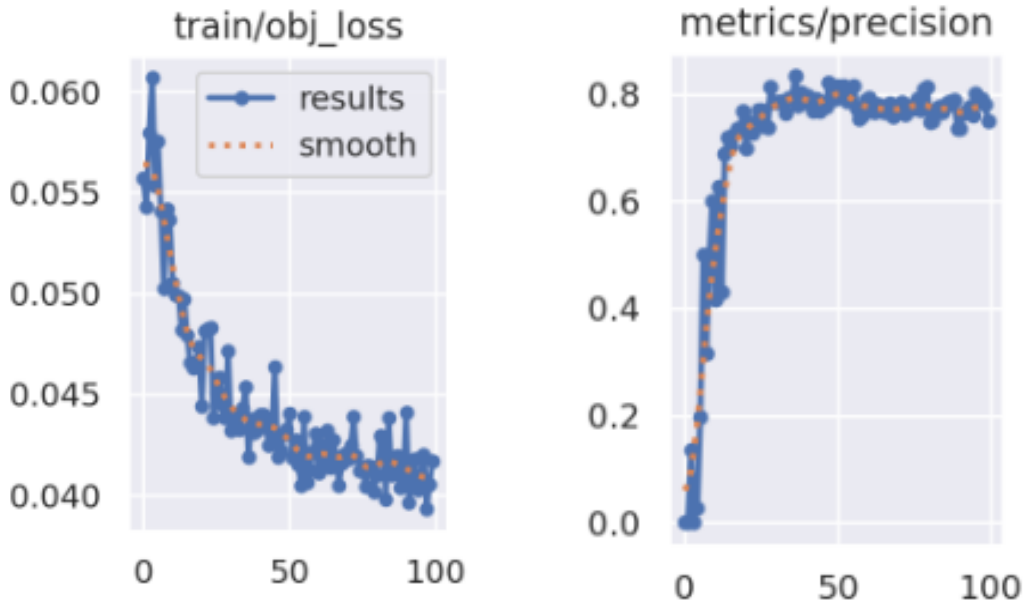



**Input:** Preprocessed Images

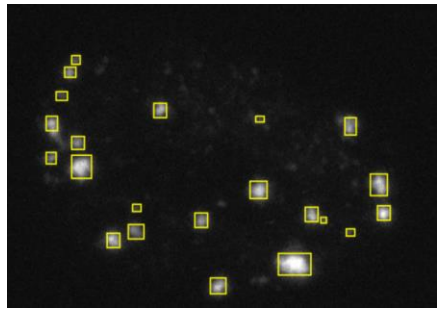
1) **Backbone:** feature extraction
2) **Neck:** feature fusion
3) **Head:** final processing to generate a model

**Output:** detected results (class, score, location, size)

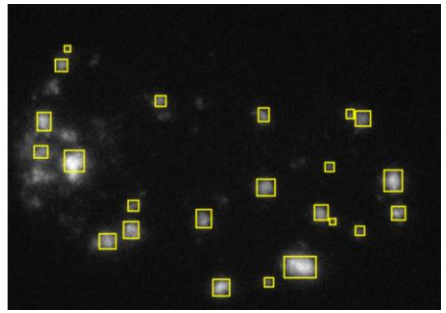
4) Using the annotated data (individual cell images with labeled Foci) as the Train/Test/Validation set for the Convolutional Neural Network based on Yolov5 algorithm

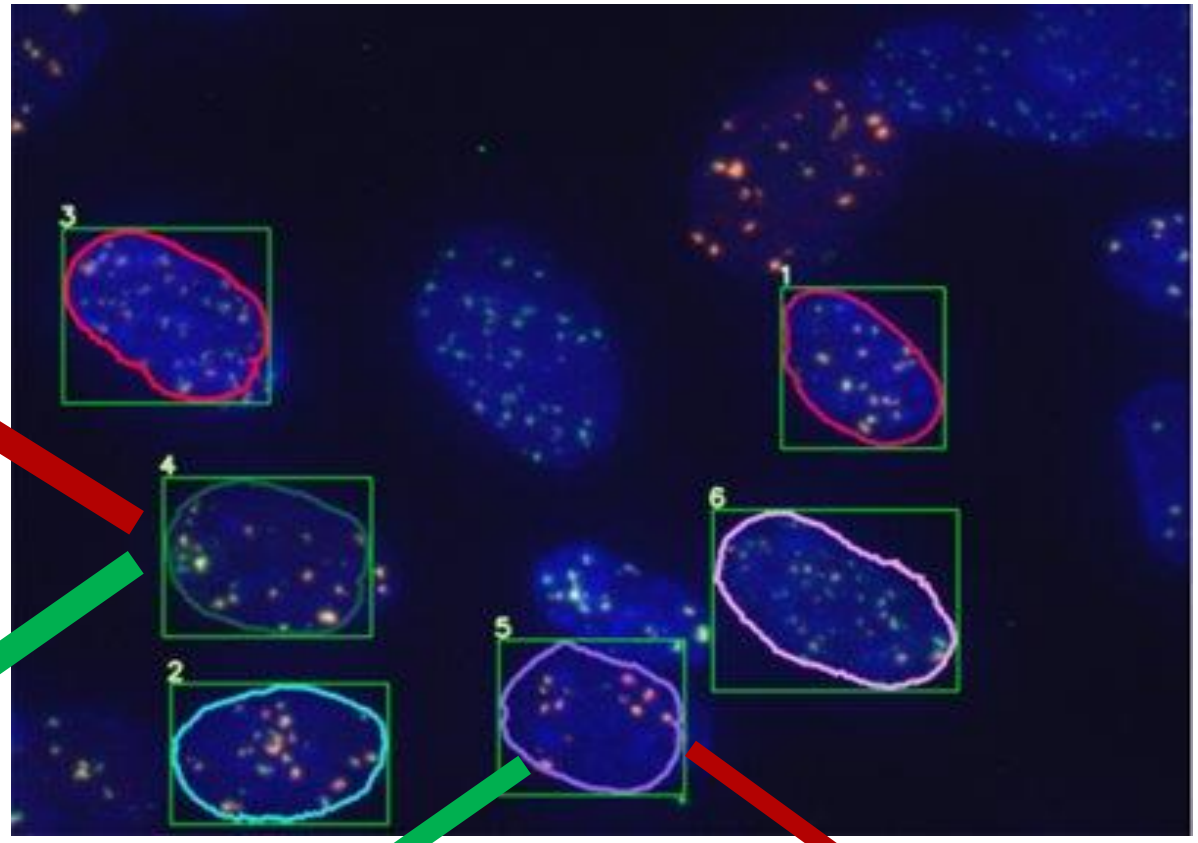


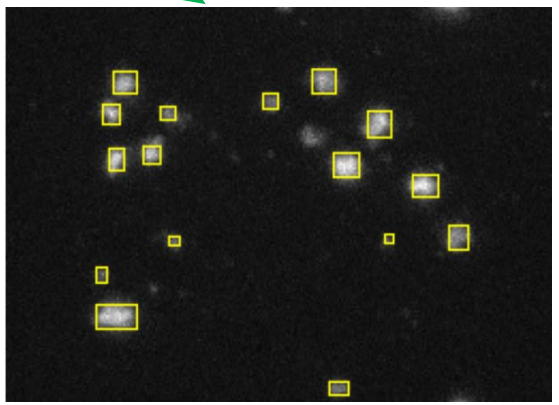
5) Applying the Trained NN on the given images

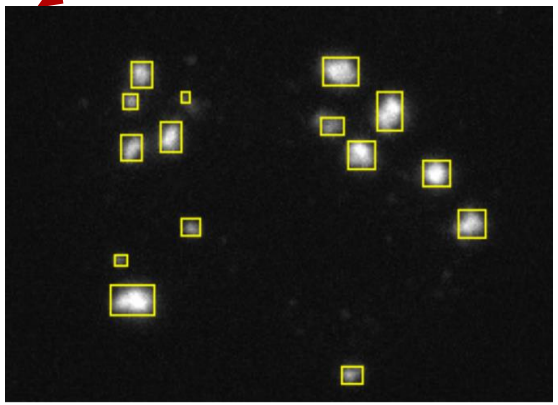


**The Detected Red and Green Foci**







# Comparison with DARFI

## Red Foci

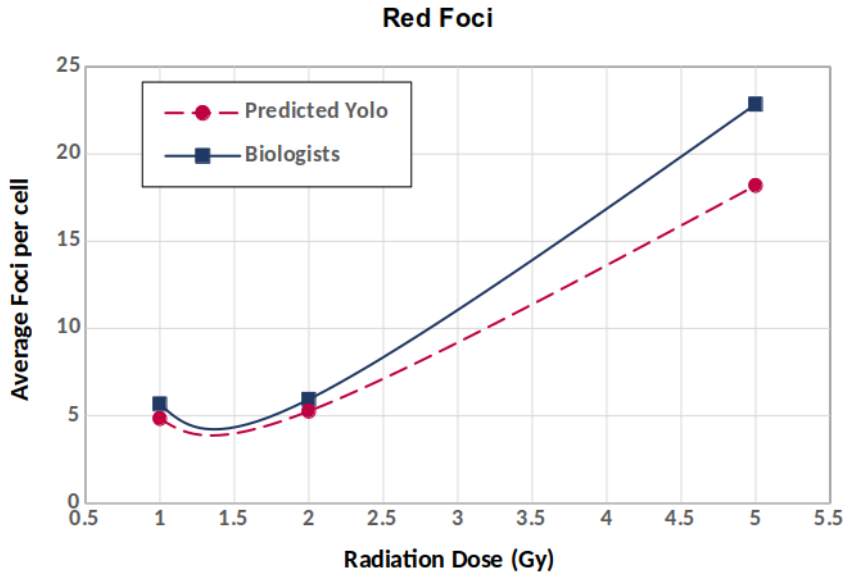
| Fibroblast , 24h after Irradiation of 1Gy | | |
|---|---|---|
| Cell_name | YOLO Predicted | DARFI |
| RedFoci_141207_0 | 12 | 13 |
| RedFoci_141207_1 | 33 | 38 |
| RedFoci_141207_2 | 15 | |
| RedFoci_141351_0 | 2 | 4 |
| RedFoci_141351_1 | 6 | 11 |
| RedFoci_141525_0 | 43 | 48 |
| RedFoci_141525_1 | 24 | 27 |
| RedFoci_141617_0 | 12 | 10 |
| RedFoci_141617_1 | 23 | 39 |
| RedFoci_142021_0 | 13 | 17 |
| RedFoci_142021_1 | 9 | 12 |
| RedFoci_142143_0 | 5 | 7 |
| RedFoci_142143_1 | 1 | 3 |

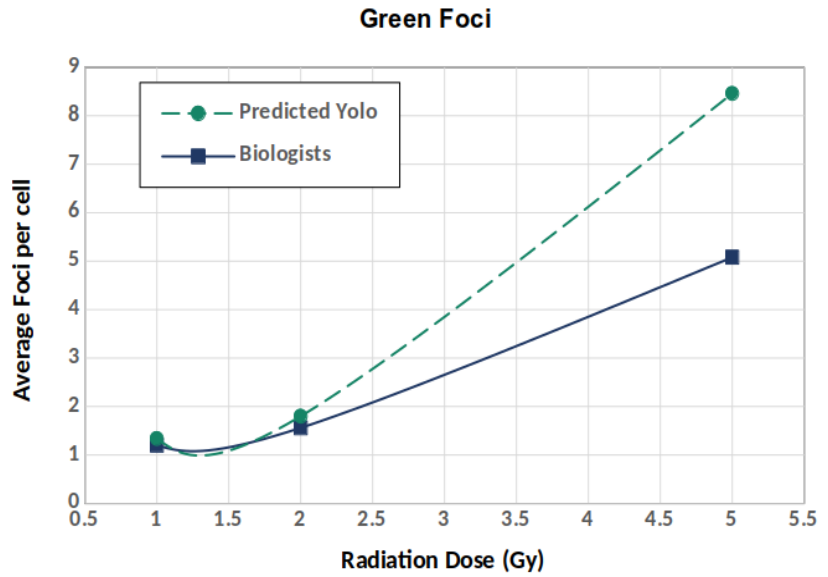
## Green Foci

| Fibroblast , 24h after Irradiation of 1Gy | | |
|---|---|---|
| Cell_name | YOLO Predicted | DARFI |
| GreenFoci_141207_0 | 3 | 4 |
| GreenFoci_141207_1 | 1 | 1 |
| GreenFoci_141207_2 | 1 | |
| GreenFoci_141351_0 | 0 | 2 |
| GreenFoci_141351_1 | 2 | 3 |
| GreenFoci_141525_0. | 5 | 4 |
| GreenFoci_141525_1 | 7 | 8 |
| GreenFoci_141617_0 | 2 | 1 |
| GreenFoci_141617_1 | 7 | 5 |
| GreenFoci_142021_0 | 1 | 3 |
| GreenFoci_142021_1 | 5 | 3 |
| GreenFoci_142143_0 | 0 | 0 |
| GreenFoci_142143_1 | 8 | 4 |

http://github.com/varnivey/darfi

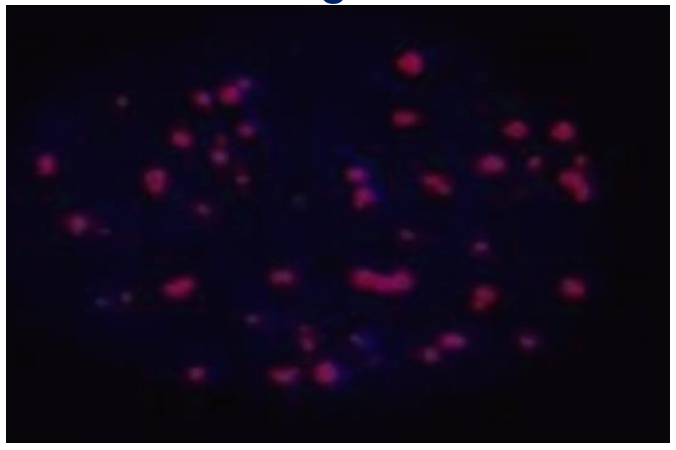
# Comparison with Biologists' Counting



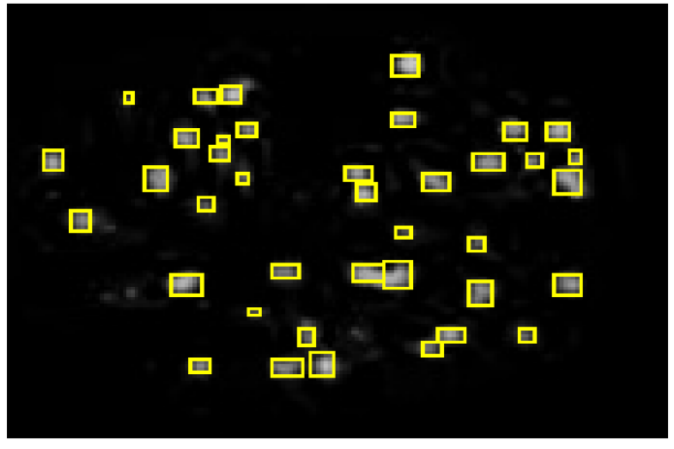



# Difficulties with DARFI

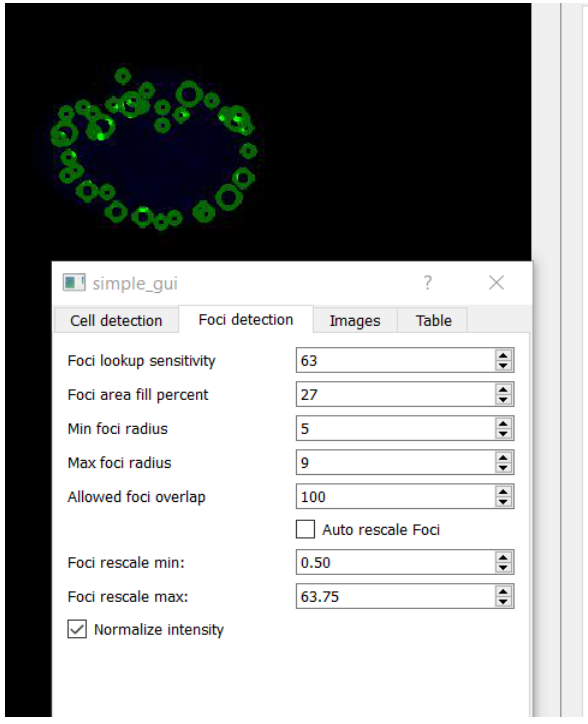
## Main Cell image with Red Foci



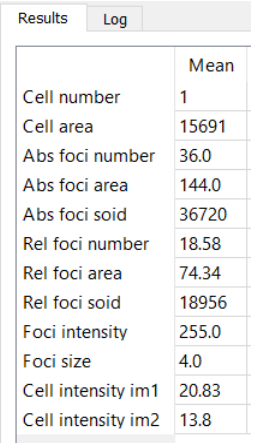
## Our Algorithm Foci Marking



DARFI _ Input Parameters 1

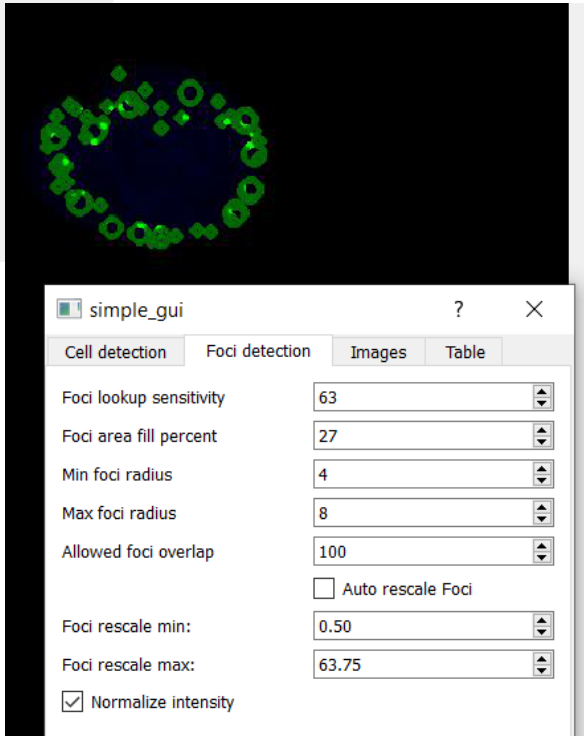



## DARFI Foci Marking



| | Mean |
|---|---|
| Cell number | 1 |
| Cell area | 15691 |
| Abs foci number | 36.0 |
| Abs foci area | 144.0 |
| Abs foci soid | 36720 |
| Rel foci number | 18.58 |
| Rel foci area | 74.34 |
| Rel foci soid | 18956 |
| Foci intensity | 255.0 |
| Foci size | 4.0 |
| Cell intensity im1 | 20.83 |
| Cell intensity im2 | 13.8 |

DARFI _ Input Parameters 2




| | Mean |
|---|---|
| Cell number | 1 |
| Cell area | 15691 |
| Abs foci number | 51.0 |
| Abs foci area | 9.0 |
| Abs foci soid | 2295 |
| Rel foci number | 26.33 |
| Rel foci area | 4.65 |
| Rel foci soid | 1185 |
| Foci intensity | 255.0 |
| Foci size | 0.18 |
| Cell intensity im1 | 20.83 |
| Cell intensity im2 | 13.8 |

# Summary and Outlook

- ✓ A deep learning algorithm is developed to automatically detect and analyze the radiation-induced foci.
- ✓ The deep learning approach consisted of two stages; a computer vision algorithm and a neural network were used to extract the cells from each image, then a deep learning model based on a Yolo algorithm was used to count the number of foci.
- ✓ The required data for training the Yolo model was annotated on the CVAT platform.

  - Adding more annotated data to the training set in order to improve the results.
  - Meanwhile we are developing a web service to automatically analyze the RIF formation.
  - The webservice will be designed to work with minimum number of input parameters.
  - It will allow the user to process the group of fluorescent images and obtain analytical information including the average number of RIF per cell and RIF area.

- ***Thanks for your attention***