



UNIwersytet
Warszawski



Głębokie sieci neuronowe w analizie obrazów na przykładzie projektów realizowanych w Centrum Uczenia Maszynowego UW

Marek Bukowicki

48 Zjazd Fizyków Polskich
Gdańsk, 1-7 września, 2023



- klasyfikacja
- detekcja obiektów
- segmentacja
- segmentacja obiektów
- detekcja punktów kluczowych (pose estimation)
- detekcja z wykryciem orientacji
- trójwymiarowa detekcja
- śledzenie obiektów
- śledzenie obiektów
- wykrywanie linii
- klasyfikacja wideo
- transfer stylu
- generowanie obrazów na podstawie tekstu/kategorii
- opisywanie zdjęć
- odpowiadanie na pytania na podstawie obrazu

Podstawowe zadania wizji komputerowej



UNIWERSYTET
WARSZAWSKI

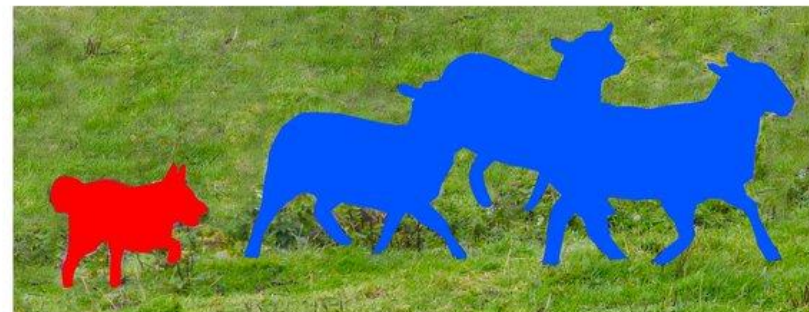


WYDZIAŁ INFORMATYKI
UCZELNIA
BADAWCZA

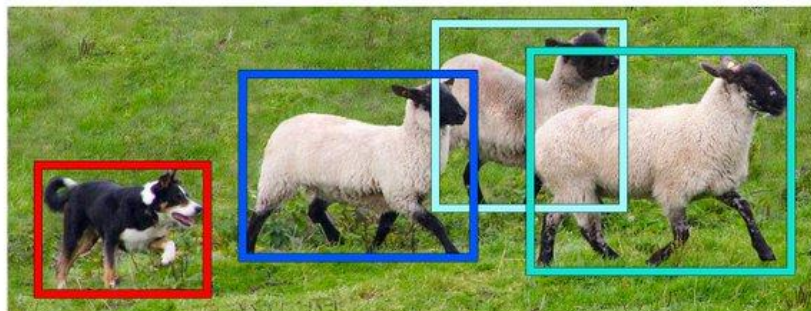
<https://center4ml.idub.uw.edu.pl>



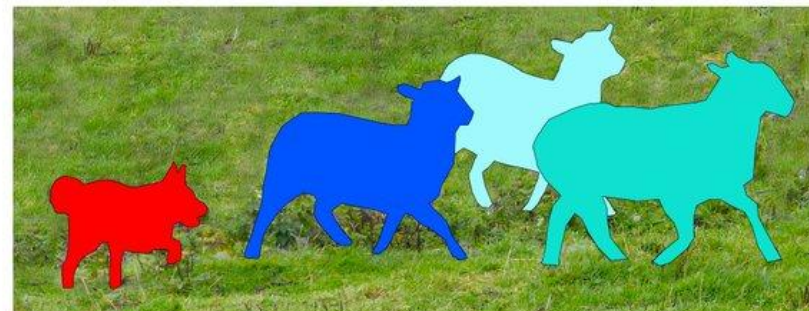
klasyfikacja **Image Recognition**



segmentacja **Semantic Segmentation**



detekcja obiektów **Object Detection**



segmentacja obiektów **Instance Segmentation**

Podstawowe zadania wizji komputerowej w nauce

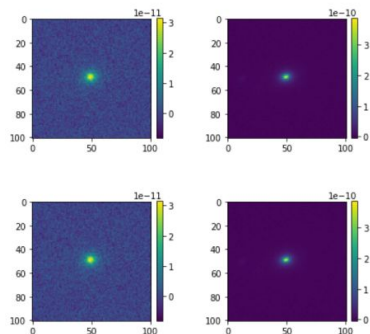


UNIwersytet
WARSZAWSKI



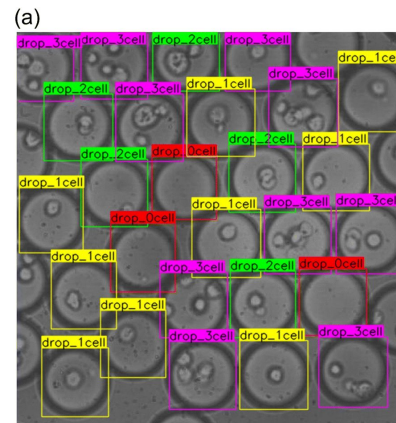
WYSTĘPIENIA I
UBIEŻNIENIA
BADAWCZA

<https://center4ml.idub.uw.edu.pl>



klasyfikacja

Thuruthipilly, H., Grespan, M., & Zadrożny, A. (2022). Transformers as Strong Lens Detectors-From Simulation to Surveys

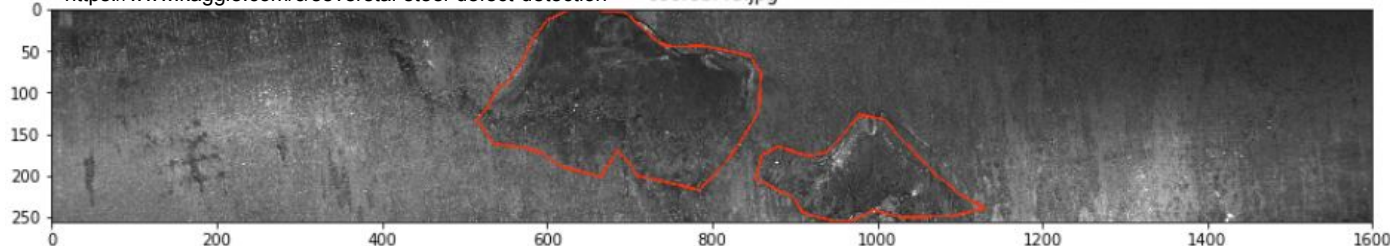


detekcja obiektów

Gardner et al. (2022). Deep learning detector for high precision monitoring of cell encapsulation statistics in microfluidic droplets.

segmentacja obiektów

<https://www.kaggle.com/c/severstal-steel-defect-detection> 000f6bf48.jpg



Narzędzia



UNIwersytet
WARSZAWSKI



INSTYTUT INFORMATYKI
UNIWERSYTETU
WARSZAWSKIEGO
UCZELNIA
BADAWCZA

<https://center4ml.idub.uw.edu.pl>

- Python!
- R
- Julia
- Matlab
- Mathematica
- ...

Narzędzia



UNIwersytet
Warszawski



Instytut Informatyki
UCZELNIA
BADAWCZA

<https://center4ml.idub.uw.edu.pl>

- Python!
- R
- Julia
- Matlab
- Mathematica
- ...

+ ChatGPT



wirtualne środowisko (venv, conda, Docker): powtarzalność + unikanie konfliktów

wersjonowanie kodu (git, porządek + archiwizacja): powtarzalność,
bezpieczeństwo

Narzędzia - Colab



- + dostęp do komputera z GPU
- + dostępny wszędzie, dobry do współpracy
- + jupyter notebook
- + izolacja środowiska - nic nie zepsuję u siebie

- zmiany w oprogramowaniu
- ograniczenia czasu uczenia
- konieczność instalowania bibliotek przy każdym uruchomieniu

Alternatywa: Amazon SageMaker

```
liczenie_komorek_coco.ipynb ☆
File Edit View Insert Runtime Tools Help Last edited on March 30

+ Code + Text

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] dataset_path = '/content/drive/MyDrive/027_kolonie_bakterii_SLCJ/02_narzedzia/edycja_oznaczen/de
output_path = '/content/drive/MyDrive/027_kolonie_bakterii_SLCJ/02_narzedzia/edycja_oznaczen/de

[ ] import pandas as pd
from pycocotools.coco import COCO

[ ] dataset = COCO(dataset_path)

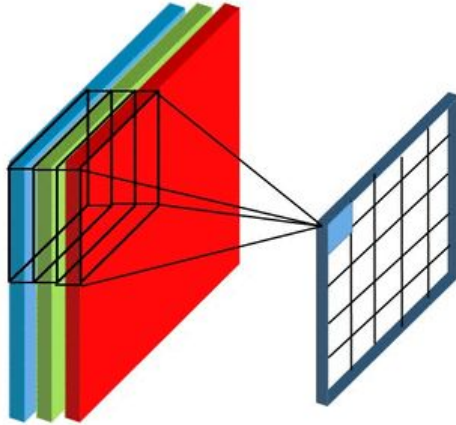
loading annotations into memory...
Done (t=0.36s)
creating index...
index created!

table_data = []
for img_id in dataset.getImgIds():
    file_name = dataset.loadImgs([img_id])[0]['file_name']
    anns = dataset.getAnnIds(imgIds=[img_id])
    N_detections = len(anns)
    table_data.append(f
```


Jak działa sieć neuronowa?



filtr: wiele kanałów → jedna wartość
(parametryzowana funkcja nieliniowa)

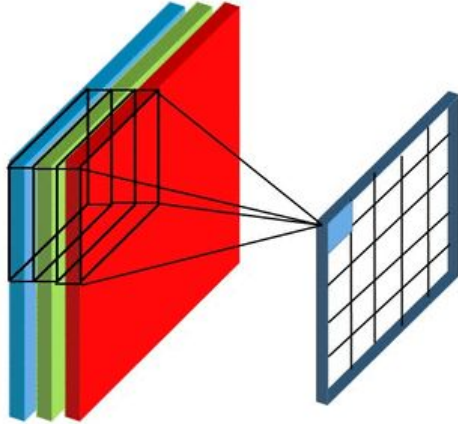


liczba parametrów (wag) = (rozmiar filtra + 1) × liczba kanałów

Jak działa sieć neuronowa?

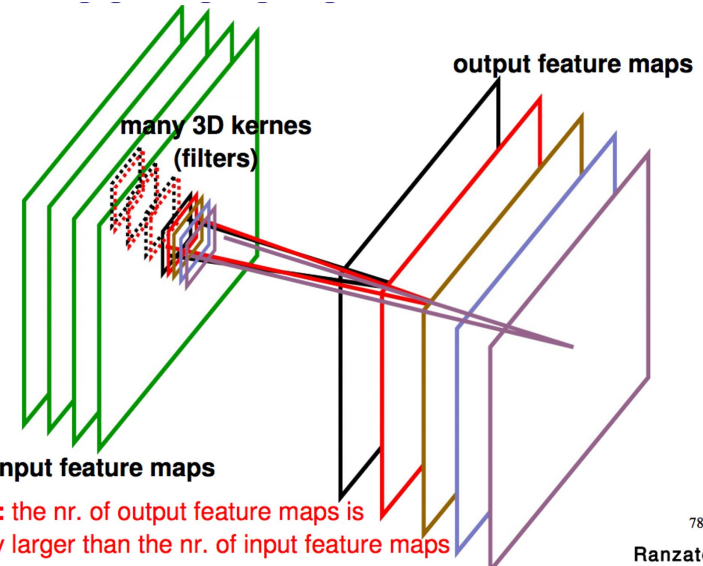


filtr: wiele kanałów → jedna wartość



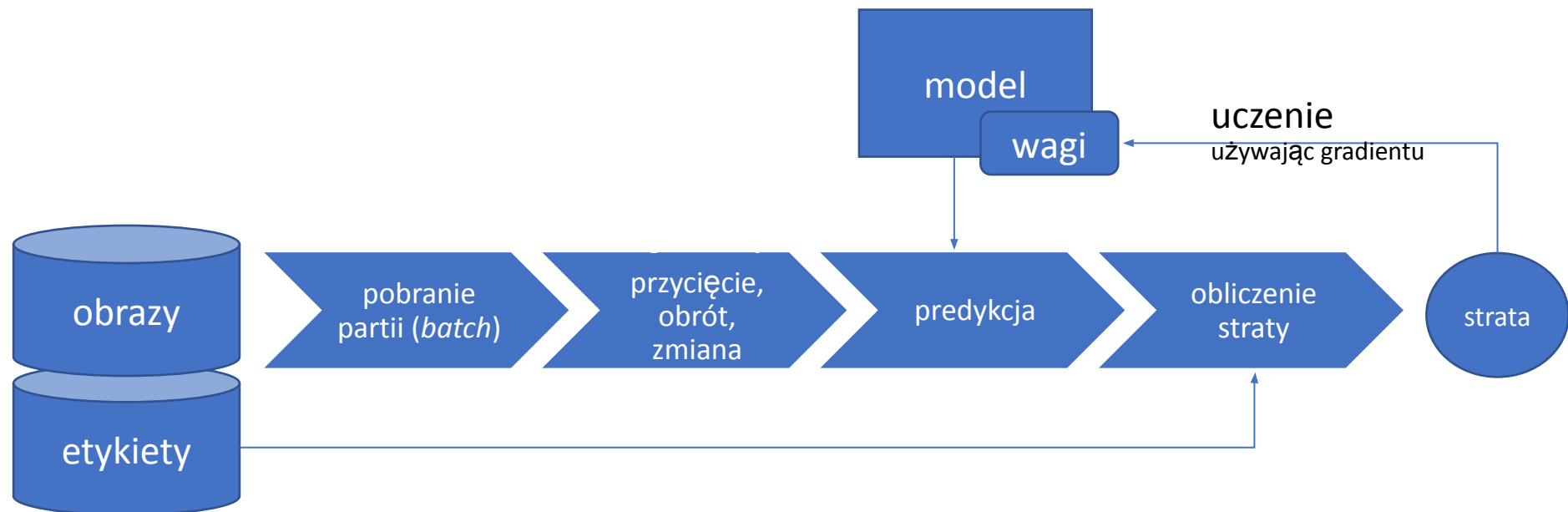
liczba wag = (rozmiar filtra + 1) × liczba kanałów

wiele filtrów → wiele kanałów wyjściowych



NOTE: the nr. of output feature maps is usually larger than the nr. of input feature maps

Jak wytrenować sieć neuronową?

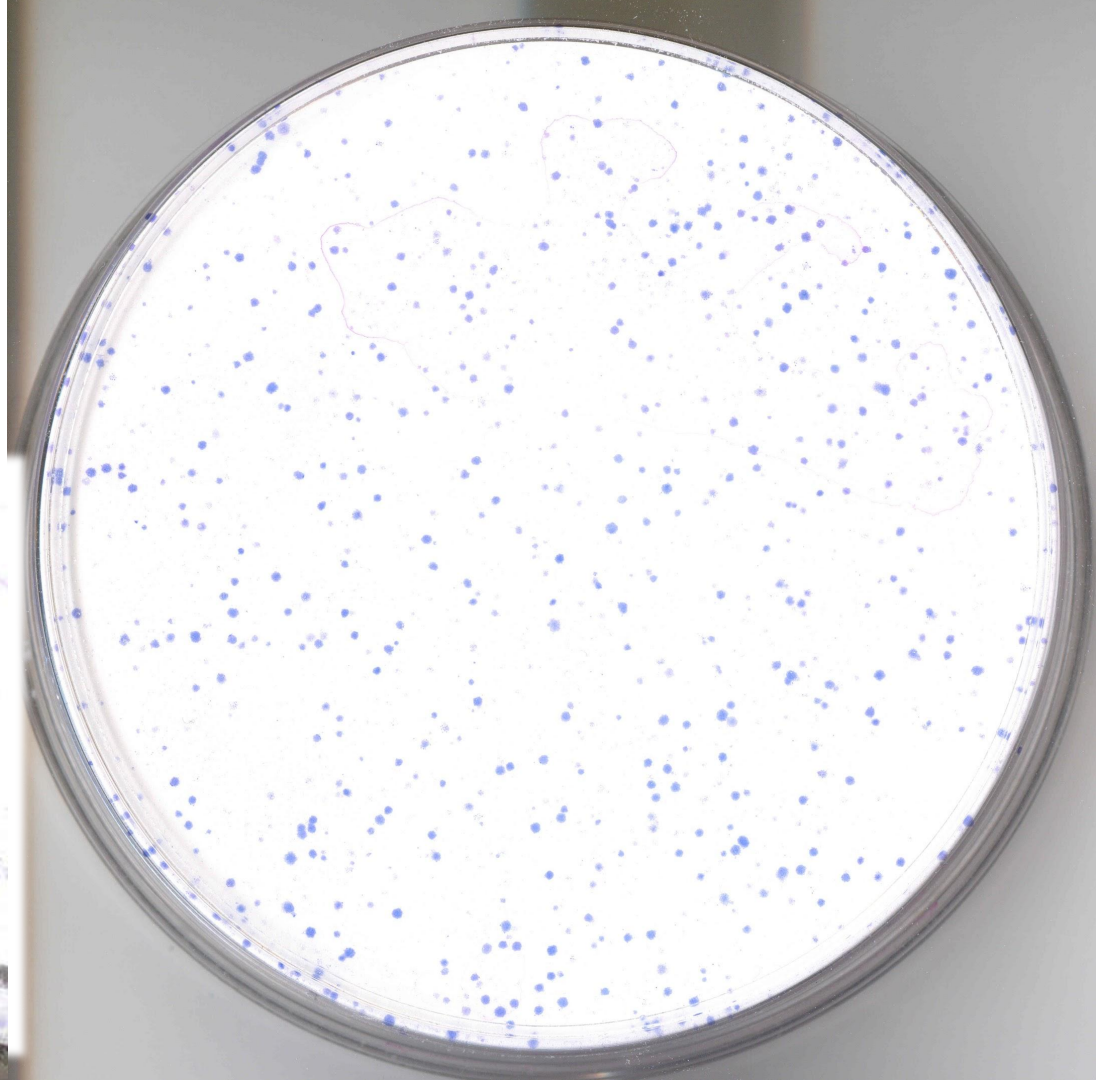
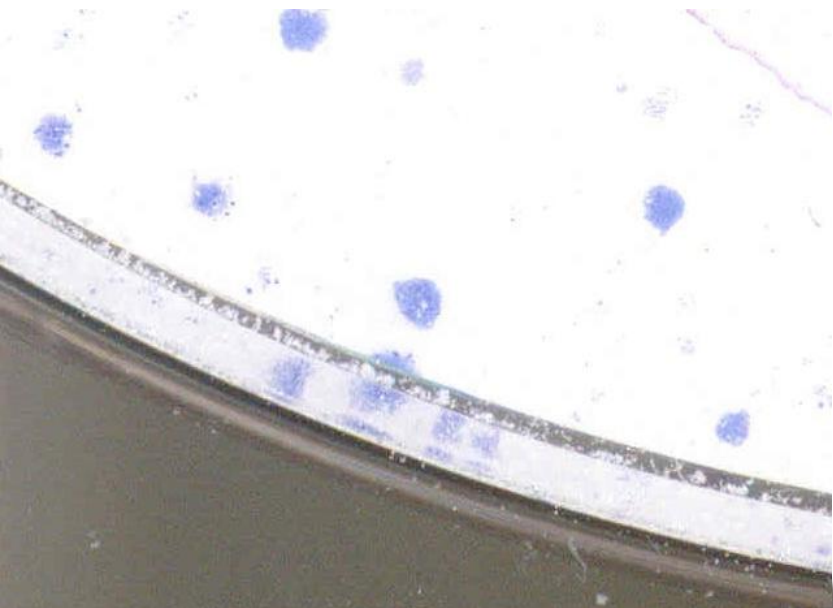


Detekcja kolonii

współpraca: ŚLCJ UW

cel: liczenie kolonii,

trudności: duże zdjęcia, niewyraźne kolonie



Dane - ile?



UNIwersytet
warszawski



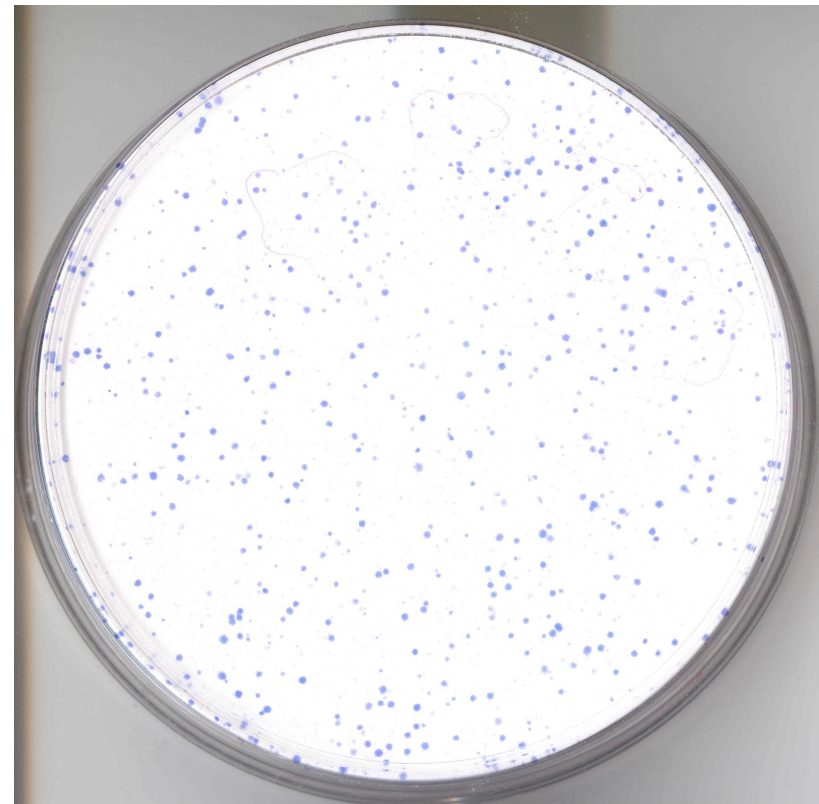
WYDZIAŁ BIOLOGII
UCZELNIA
BADAWCZA

<https://center4ml.idub.uw.edu.pl>

kilka tysięcy obiektów (kolonii)

Rekomendacje:

- min. setki obiektów dla każdej kategorii
- najlepiej tysiące
- zależy od wymaganej dokładności i cech obiektu
- czasem wystarczy ok 100



Dane - problemy



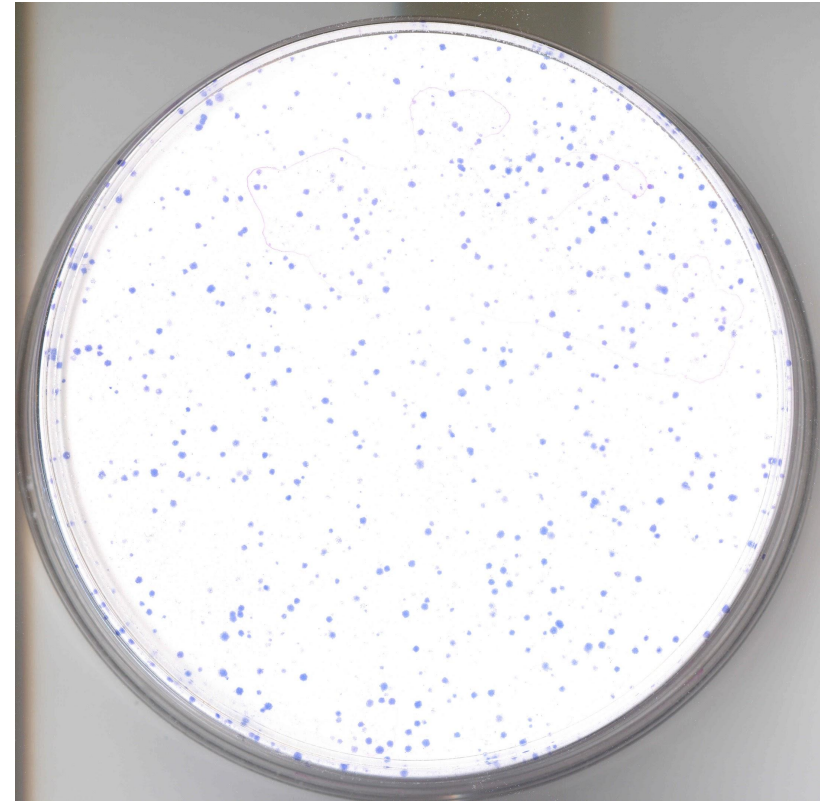
UNIwersytet
warszawski



WYDZIAŁ
UCZELNIA
BADAWCZA

<https://center4ml.idub.uw.edu.pl>

duże zdjęcia - konieczny podział



Detekcja kolonii - jak oznaczyć?

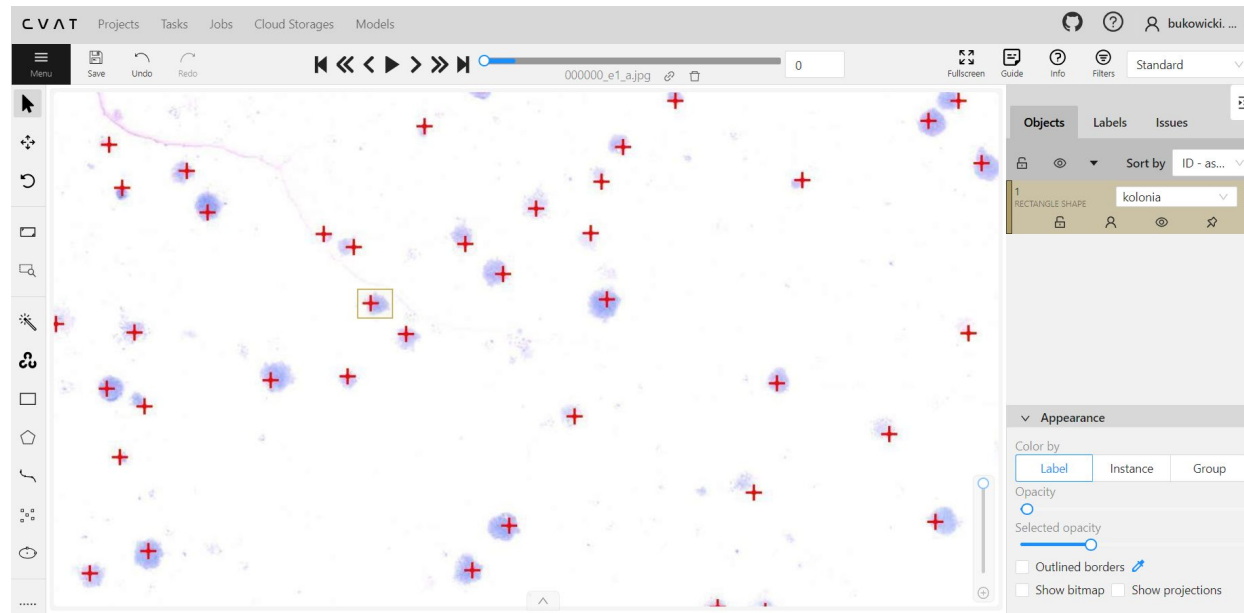


Narzędzia:

- make sense (online)
- Label Studio (lokalnie)
- Labellmg (lokalnie)
- Roboflow (online)
- CVAT (online)

Wynik: plik/pliki tekstowe
z oznaczeniami:

- kategoria
- położenie
- nazwa pliku



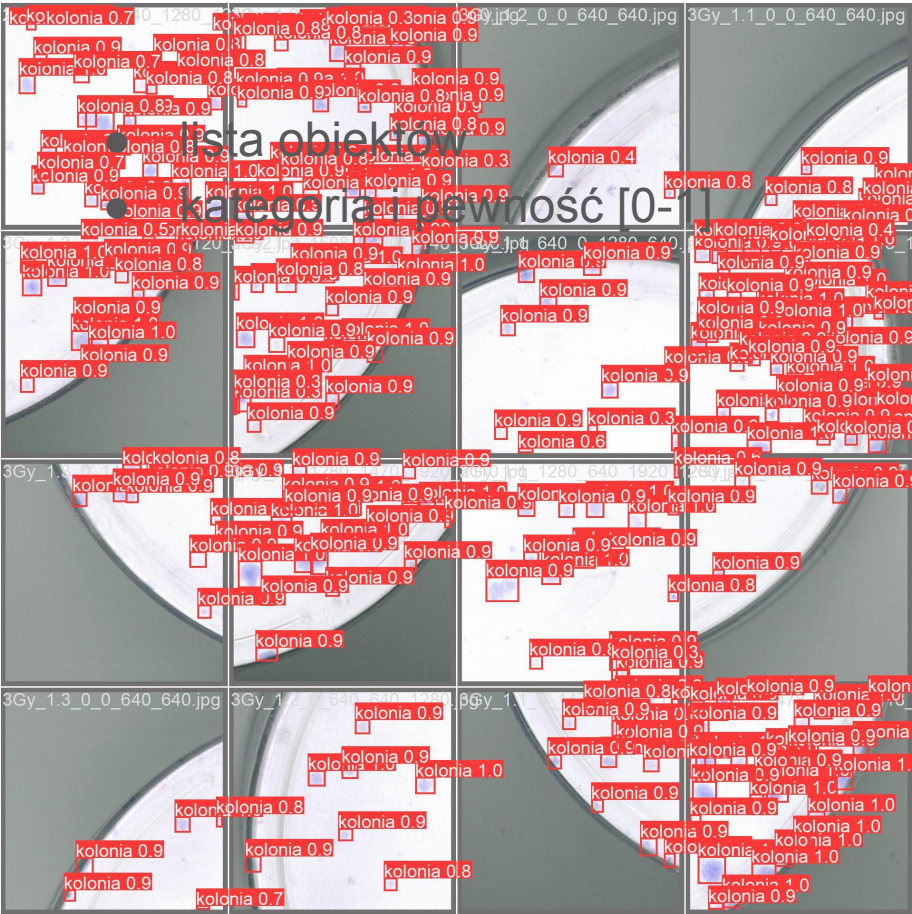

```
# Build a new model from YAML, transfer pretrained weights to it and start training
yolo detect train data=coco128.yaml model=yolov8n.yaml pretrained=yolov8n.pt epochs=100 imgsz=640
```

Supported Tasks

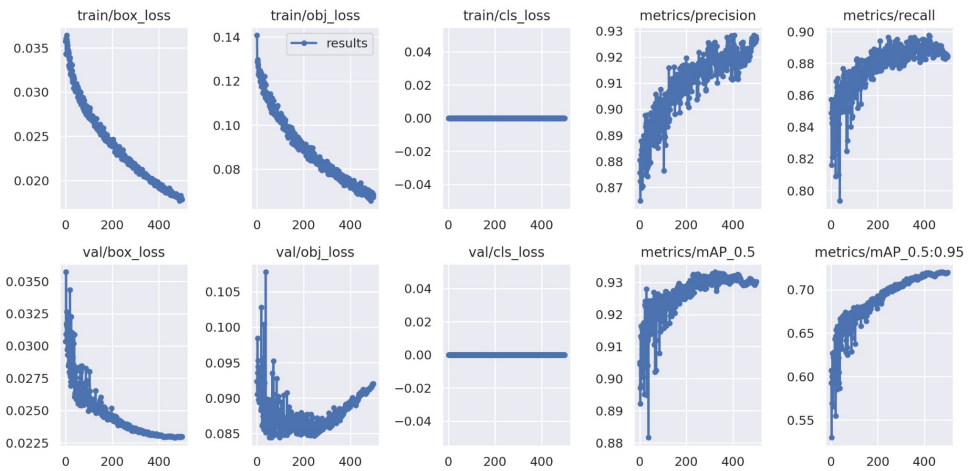
Model Type	Pre-trained Weights	Task
YOLOv8	yolov8n.pt , yolov8s.pt , yolov8m.pt , yolov8l.pt , yolov8x.pt	Detection
YOLOv8-seg	yolov8n-seg.pt , yolov8s-seg.pt , yolov8m-seg.pt , yolov8l-seg.pt , yolov8x-seg.pt	Instance Segmentation
YOLOv8-pose	yolov8n-pose.pt , yolov8s-pose.pt , yolov8m-pose.pt , yolov8l-pose.pt , yolov8x-pose.pt , yolov8x-pose-p6.pt	Pose/Keypoints
YOLOv8-cls	yolov8n-cls.pt , yolov8s-cls.pt , yolov8m-cls.pt , yolov8l-cls.pt , yolov8x-cls.pt	Classification

trening na GPU w colabie (kilka godzin)

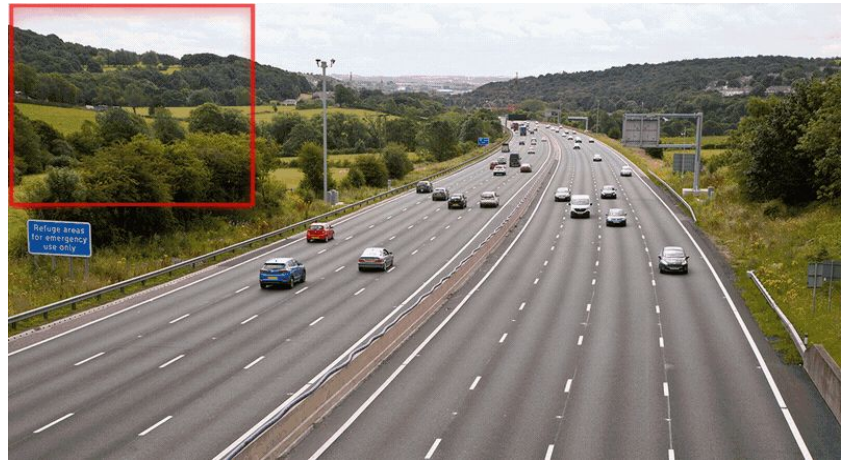
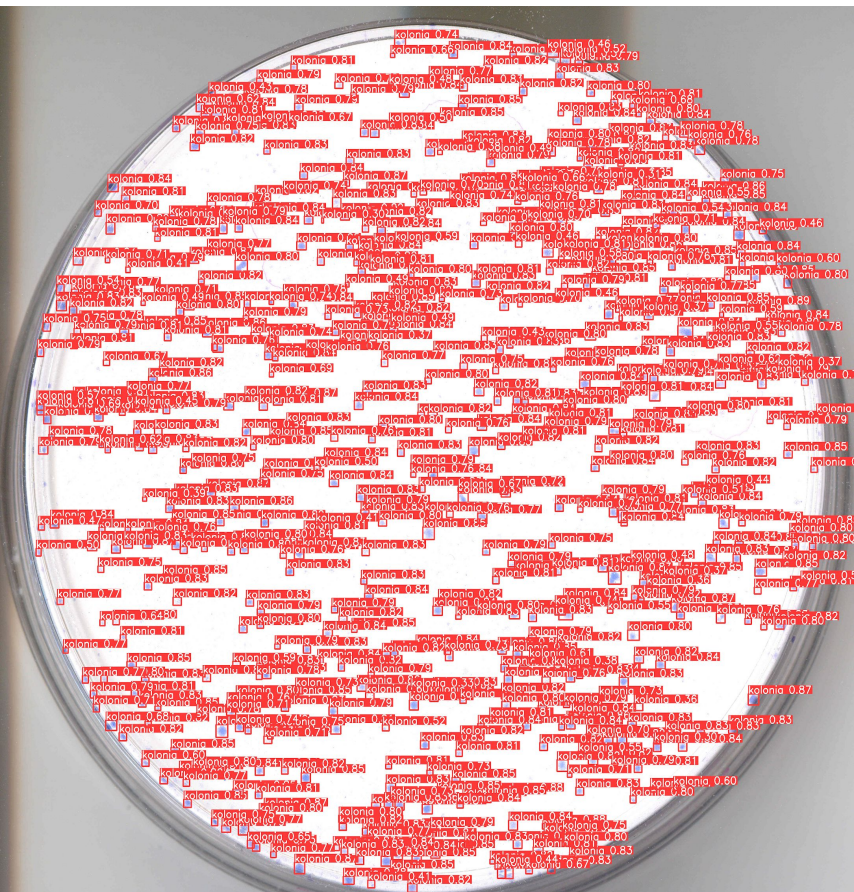
Detekcja kolonii - wyniki



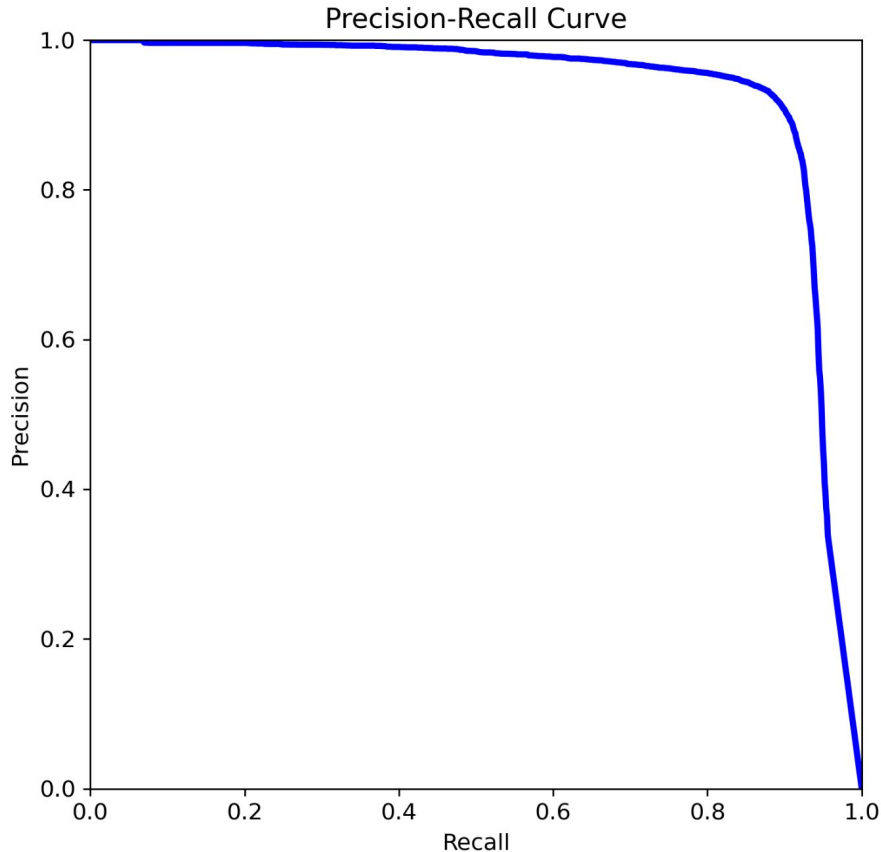
lista obiektów
kategoria: dawność [0-1]



Detekcja kolonii - wyniki



Detekcja kolonii - ustawienie progów



— kolonia 0.931
— all classes 0.931 mAP@0.5

- 0.93 mAP@0.5
- precyzja 89%, specyficzność 87%
- błąd względny 3% na nowych danych

Dane - wizualizacja



UNIwersYTET
WARSZAWSKI



<https://center4ml.idub.uw.edu.pl>

YOLO

samodzielnie

Voxel51

Copy of Result_exploration_voxel51_v2.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

FiftyOne kolonie + add stage

RAM Disk

Editing

Comment Share

Sample tags Label tags Labels Other fields

Filter

SAMPLE TAGS

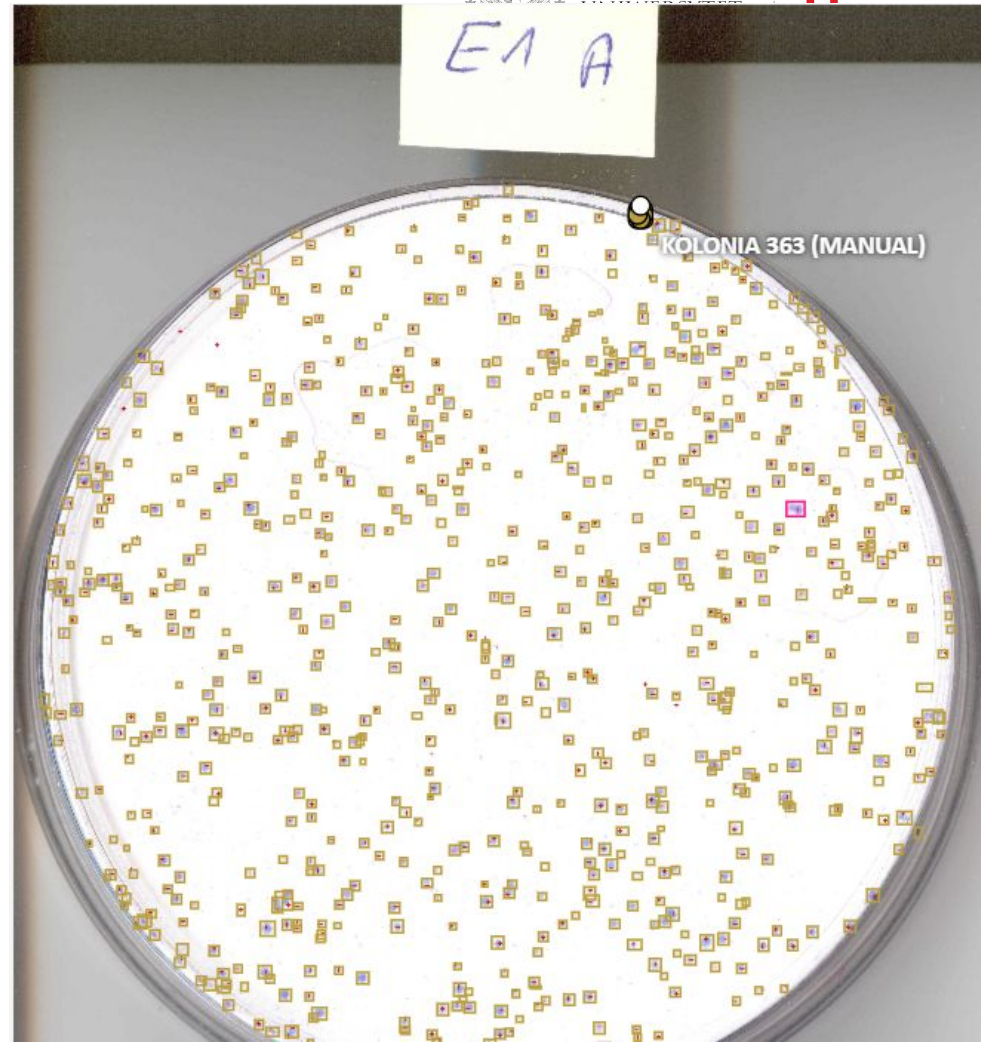
- folder_14Gy/e1_a.jpg 1
- folder_14Gy/e1_b.jpg 1
- folder_14Gy/e1_c.jpg 1
- folder_14Gy/e2_a.jpg 1
- folder_14Gy/e2_b.jpg 1
- folder_14Gy/e2_c.jpg 1
- folder_14Gy/e3_a.jpg 1
- folder_14Gy/e3_b.jpg 1
- folder_14Gy/e3_c.jpg 1
- folder_14Gy/e5_a.jpg 1
- folder_14Gy/e5_b.jpg 1
- folder_14Gy/e5_c.jpg 1

56 samples

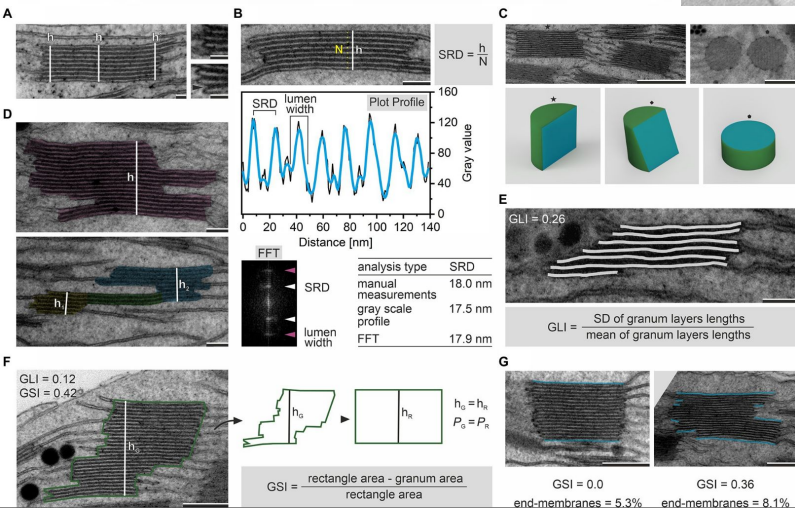
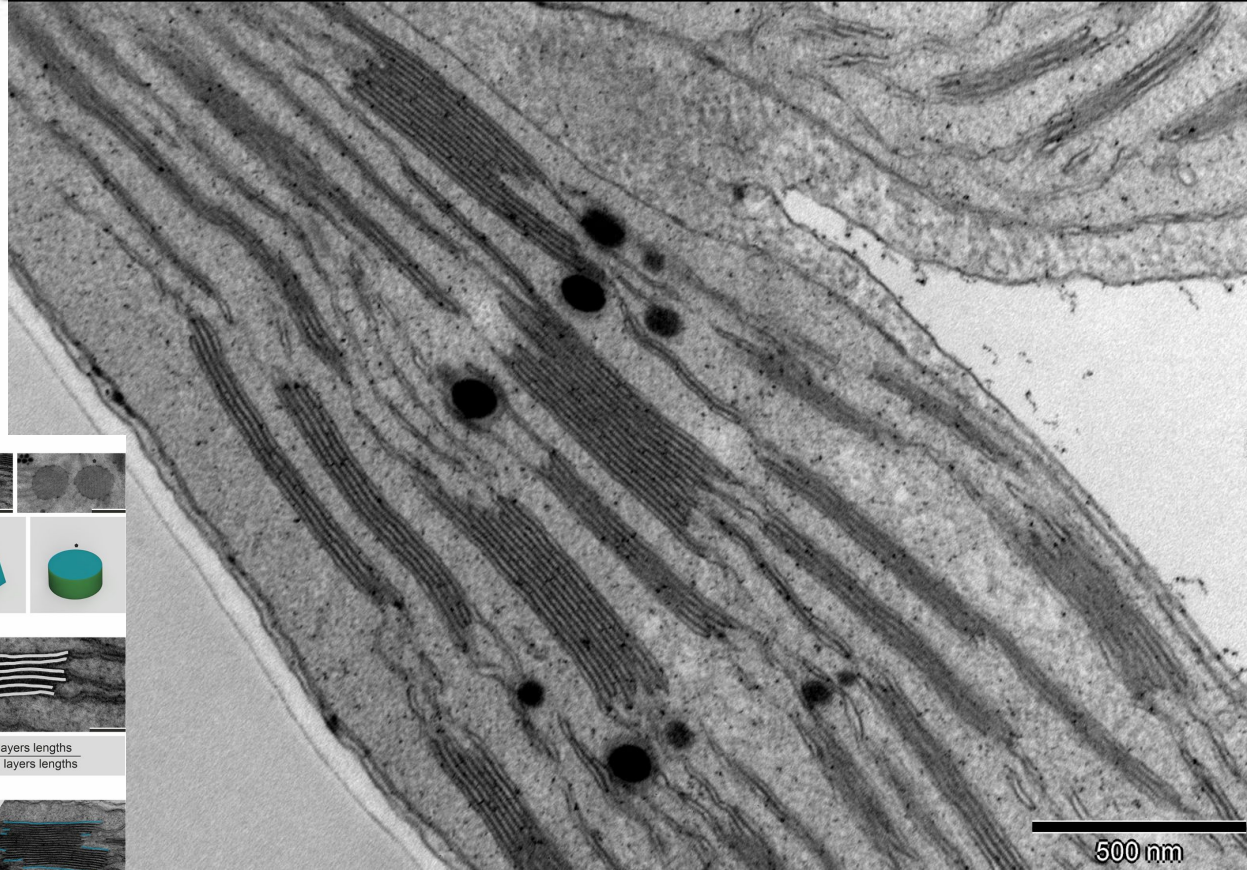
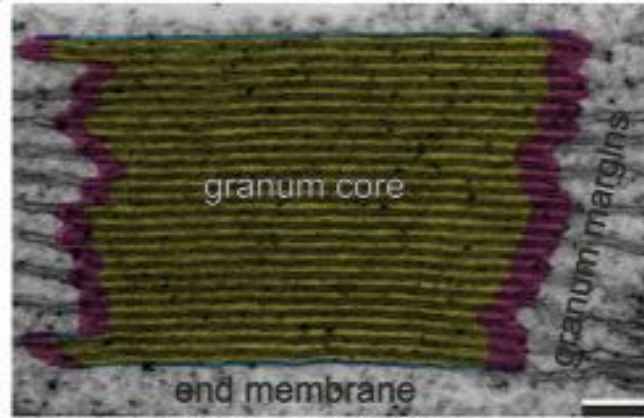
0s completed at 4:10 PM

Douczenie

- oznaczanie (edycja predykcji)
- podział zdjęć
- uczenie
- ewaluacja



Pomiar granum



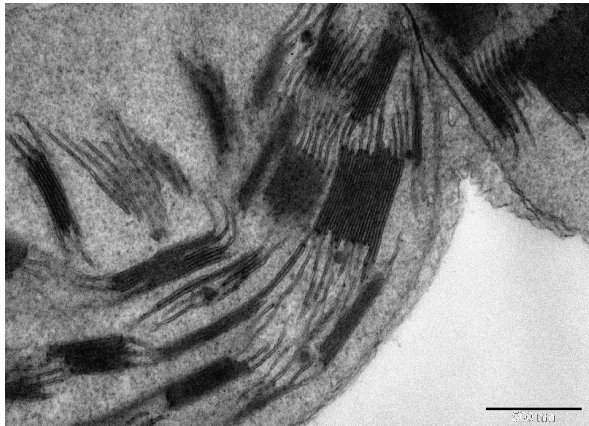
500 nm

Pomiar granum

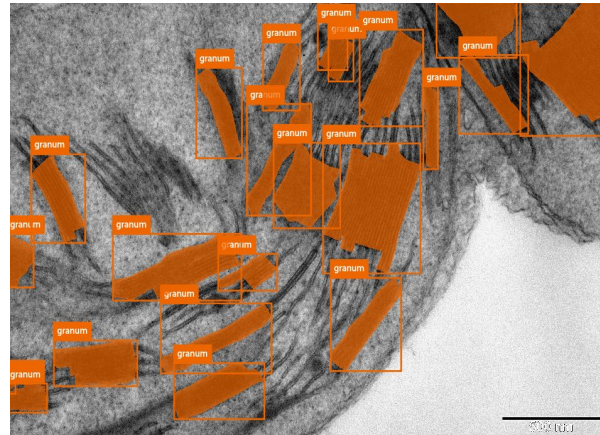


segmentacja obiektów

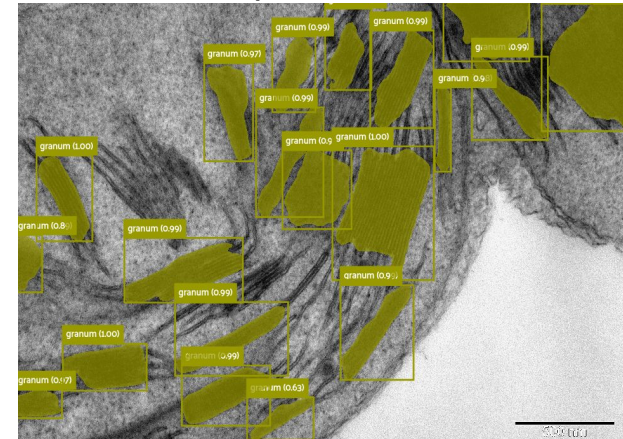
original images



labels



prediction

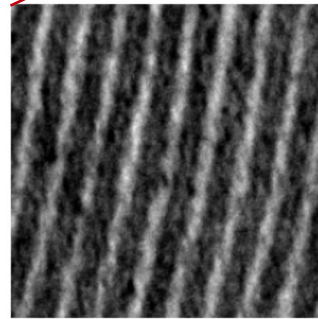


Pomiar granum

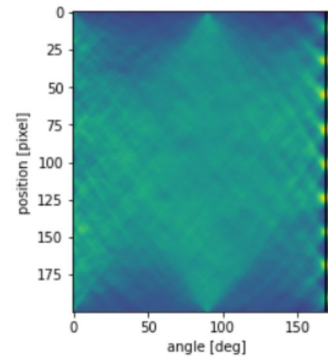


segmentacja obiektów

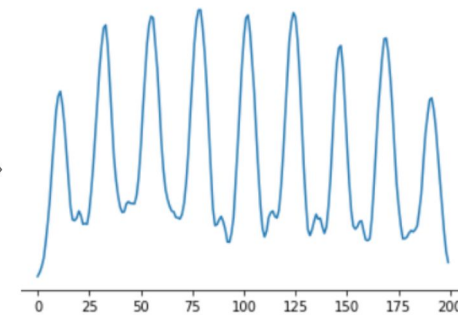
+ klasyczne metody CV



original image crop

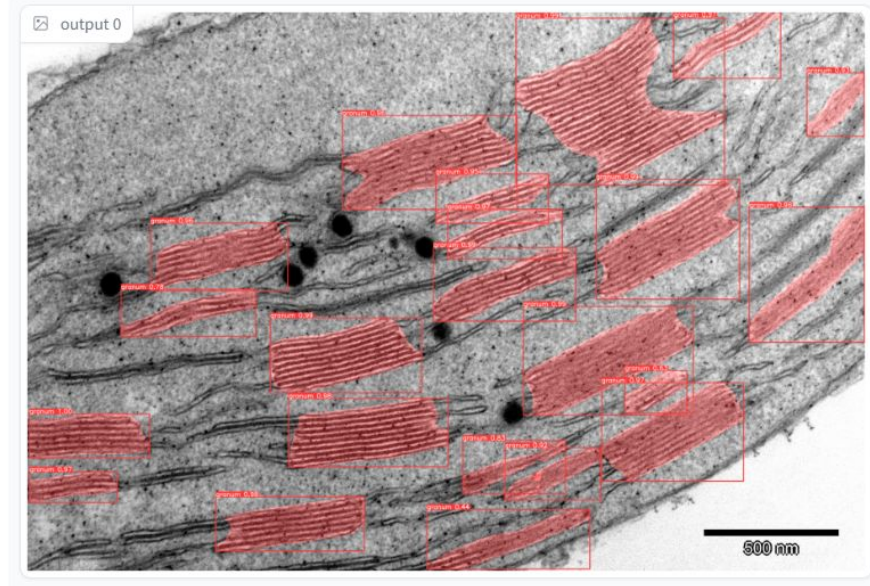
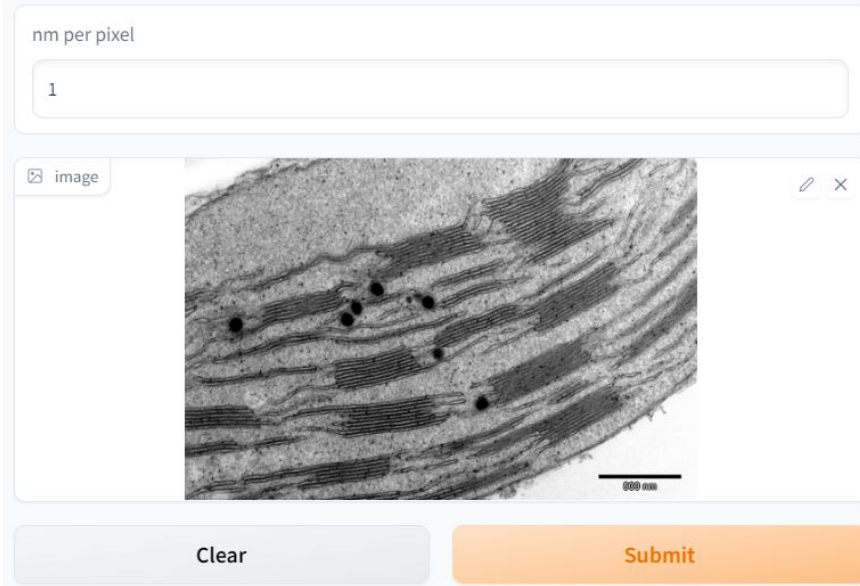


orientation detection



period detection

Pomiar granum - interfejs



output 1

idx	period	period_quality	white_strip_width	N layers
0	10	0.7555860868097886	3.076923076923077	16
1	13.5	0.8344871111303597	4.038461538461538	15
2	25	0.9478899925247282	13.8	12
3	27	0.745280371569983	14.707317073170731	21

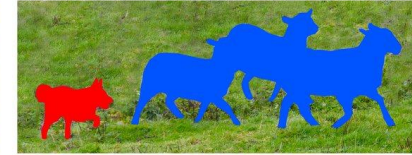


interaktywne API
(pojedyncze obrazy)
+
kod na github-ie

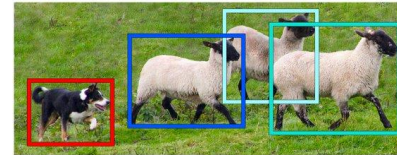
- cel:
 - zwiększenie skali badań
 - zmniejszenie nakładu pracy
- dojrzałe zagadnienia:
 - klasyfikacja
 - segmentacja
 - detekcja i segmentacja obiektów
 - detekcja punktów kluczowych
 - śledzenie obiektów
- podstawowe modele są proste do zastosowania (potrzebne podstawy programowania)
- uczenie lokalnie (potrzebne GPU) lub w chmurze
- użyteczne biblioteki: YOLO / Detectron2 / MMDetection / pytorch lightning



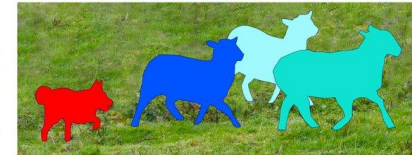
Image Recognition



Semantic Segmentation



Object Detection



Instance Segmentation



UNIwersytet
Warszawski



Instytut
Uczelnia
Badawcza

<https://center4ml.idub.uw.edu.pl>

Dziękuję za uwagę