Технически университет - София

Курсов проект

Технологии за големи данни

“Разпознаване на бактерии в петри чинийки и сегментация на локацията на бактерията”

Разработили:

Йоанна Благоева 961324010, АГМПД 2024/25

Таня Узунова 961324004, АГМПД 2024/25

Въведение

Съвременните технологии за машинно обучение са ключови за автоматизацията и оптимизацията на процеси в различни индустрии. В този проект разглеждаме разработката на модел за разпознаване на обекти в изображения, като по-конкретно бактериални колонии на лабораторни проби и петри чинии, използвайки предварително анотирани данни във формат COCO. Това включва използване на модерни библиотеки и техники за машинно обучение с цел създаване на ефективна система за класификация и анализ.

Литературно проучване

Приложение на ResNet в класификацията на секвенции

В микробиологията, колонообразуващите единици (CFU - colony-forming unit) се използват за определяне на броя на микробни клетки, които са жизнеспособни да се делят чрез бинарно деление в контролирани условия, като петрита за седиментни частици или контактни петрита, които се развиват при определена температура за определен период от време. В клиничната микробиология, хранителната индустрия и в проучванията за ваксини, бройката на колонообразуващите единици е от съществено значение за определяне на качеството и контрола на средата [1].

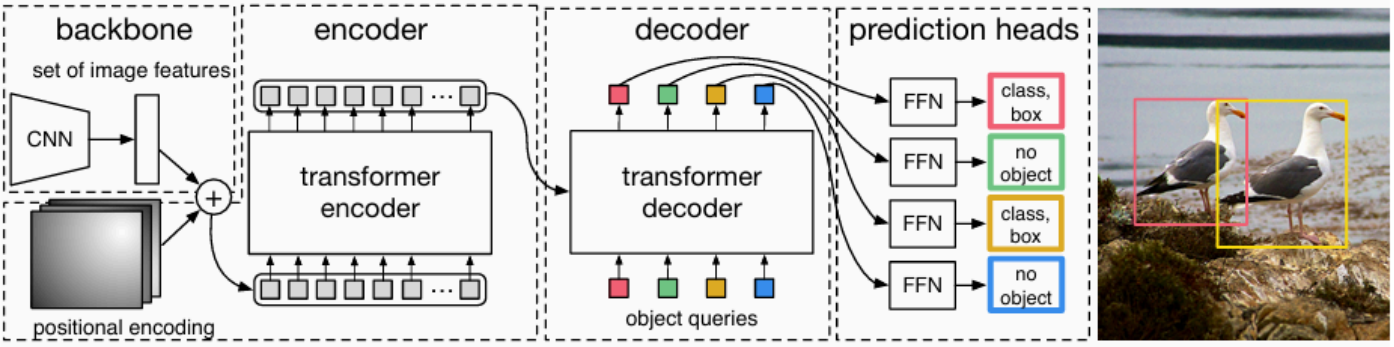
Количественото определяне на броя колонии е ключово, но процесът на броене е едновременно времеемък и трудово-интензивен. Разработвани са набори от инструменти (EBImage, ImageJ, OpenCFU, AutoCellSeg, CFUCounter), които по зададен критерий (пр. “цвят”) се използват за автоматизирано броене на обектите [2, 3, 4, 5, 6]. Въпреки широкото използване на тези решения в лабораторни условия, те са податливи откъм грешно идентифициране на колонните, като въздушни балончета или повърхностни замърсители в средата или стената от петри чинийката.

В последните години, за автоматизирането на процеса за броене се разглеждат AI и deep-learning тренировъчни модели [7]. Съвременните deep

learning object detection техники се разделят на две категории: едноетапни и двуетапни обектно разпознаващи алгоритми. Едноетапните алгоритми отгатват класовете и генерират очертаващи кутии (bounding boxes), примери за това са: You Only Look Once (YOLO) модела и Single Shot Multi-Box Detector (SSD) [8, 9]. Двуетапните модели разпознават обекта в два процеса - първия генерира предположението, което впоследствие бива класифицирано, и финно-настройва (fine-tunes) предположението, за да постигне идентифициране на обекта и неговото локализиране [7].

Навлизащите Transformer модели подновяват множество сфери в машинното обучение, включително и разпознаването на обекти. Такъв трансформър е на Facebook AI end-to-end Transformer (DETR) архитектура, която използва конволюционната невронна мрежа с архитектура ResNet-50, за да изучи 2D представяне на входящи изображения [8, 9].

DETR архитектурата описана във Фигура 1, съдържа три основни компонента: CNN основа за да извлече уплътнено свойнствено представяне, енкодер-декодер трансформатор и feed forward network (FFN), който прави финалното предположение [8].



Фиг. 1. Архитектура на DETR модел с основа ResNet-50

1. Цел на изследването

Целта на изследването е да се разработи ефективен и евтин модел за разпознаване и класификация на бактериални колонии в изображения на

петри чинийки, който може да идентифицира различни видовете бактериални колонии в лабораторна среда и да бъде широко спектърен с лесна имплементация в научната сфера. Основните задачи включват обработка и анализ на изображения, използване на предварително обучени трансформер модели, като DETR, и оптимизация на представянето на модела.

2. Експериментална рамка – набори данни (datasets), избор на метод и техника за анализ, използвани библиотеки и софтуерни средства

2.1. Набори данни (Datasets)

Dataset-а използван тук е Face-bacteria-coco. Наборът от данни включва 2527 изображения на лабораторни проби, категоризирани в следните групи: PetriDish, CompactDryEC, E-Coli, P-Aeruginosa и S-Aureus. Изображенията са анотирани във формат COCO, като всеки обект е представен с bounding boxes и съответната категория.

Данните са предварително обработени с техники като преоразмеряване до 640x640 пиксела, промяна на експонацията и прилагане на Gaussian blur.Тези изображения са разделени на тренировъчен набор от 2068 изображения и валиден набор от 286 изображения. На Фигура 2 е показано примерно изображение от избрания набор от данни.

Фиг. 2. Изображение на клас PetriDish с bounding box от клас P-Aeruginosa

2.2. Избор на метод и техника за анализ

За анализ се използва трансформерния модел DETR (DEtection TRansformer), който прилага конволюционната невронна мрежа с архитектура ResNet-50, който съчетава предимствата на невронни мрежи и трансформерите за разпознаване на обекти [8, 9]. DETR позволява ефективна обработка на анотации и предлага висока точност при класификация и локализация на обектите. Основните настройки на модела включват: използване на AdamW оптимизатор с начална скорост на обучение 1e-4 за класификационните слоеве и 1e-5 за основа на модела, комбинирана загуба, включваща cross-entropy и L1 loss, и обучение за 30 епохи с регуларизация на градиентите.

2.3. Използвани библиотеки и софтуерни средства

В рамките на представения труд бяха използвани PyTorch и torchvision за дефиниране и обучение на модела, Hugging face Transformers за интеграция с DETR, PyTorch Lightning за управление на тренировъчния процес и Matplotlib за визуализация на резултатите от модела [10, 11, 12].

3. Обработка и анализ на данните

3.1. На Фигура 3 е представен класа за обработка на данни, който е специално модифициран за съвместяване и използване с DETR. Той комбинира стандартната функционалност на CocoDetection с допълнителни стъпки за обработка на изображенията и анотациите. За целта се използва DetrImageProcessor за нормализация и преобразуване на анотациите в подходящ формат за DETR.

```
class CocoDetection(torchvision.datasets.CocoDetection):
    def __getitem__(self, idx):
        img, target = super(CocoDetection, self).__getitem__(idx)
        encoding = self.processor(images=img, annotations=target, return_tensors="pt")
        pixel_values = encoding["pixel_values"].squeeze()
        target = encoding["labels"][0]
        return pixel_values, target
```

Фиг. 3. Клас на обработка на данните

3.2. Данните се подават на модела чрез DataLoader, който пакетира изображенията и анотациите в удобен за обучение формат. Функцията collate_fn, показана на Фигура 4, осигурява консистентност в подреждането на данните и подготовката им за модела.

```
def collate_fn(batch):
    pixel_values = [item[0] for item in batch]
    encoding = processor.pad(pixel_values, return_tensors="pt")
    labels = [item[1] for item in batch]
    return {
        'pixel_values': encoding['pixel_values'],
        'pixel_mask': encoding['pixel_mask'],
        'labels': labels
    }
```

Фиг. 4. Съдържание на функцията collate_fn()

4. Представяне и визуализация на резултатите

4.1. Резултатите от модела се визуализират чрез Matplotlib. Изображенията се показват с наложени bounding boxes (очертаващи контури), етикети и точности, които са извлечени от резултатите на модела. Имплементацията е показа във фигура 5.

```
def plot_results(pil_img, scores, labels, boxes):
    plt.figure(figsize=(16,10))
    plt.imshow(pil_img)
    ax = plt.gca()
    colors = COLORS * 100
    for score, label, (xmin, ymin, xmax, ymax),c  in zip(scores.tolist(), labels.tolist
(), boxes.tolist(), colors):
        ax.add_patch(plt.Rectangle((xmin, ymin), xmax - xmin, ymax - ymin,
                                   fill=False, color=c, linewidth=3))
        text = f'{id2label[label]}: {score:0.2f}'
        ax.text(xmin, ymin, text, fontsize=15,
                bbox=dict(facecolor='yellow', alpha=0.5))
    plt.axis('off')
    plt.show()
```

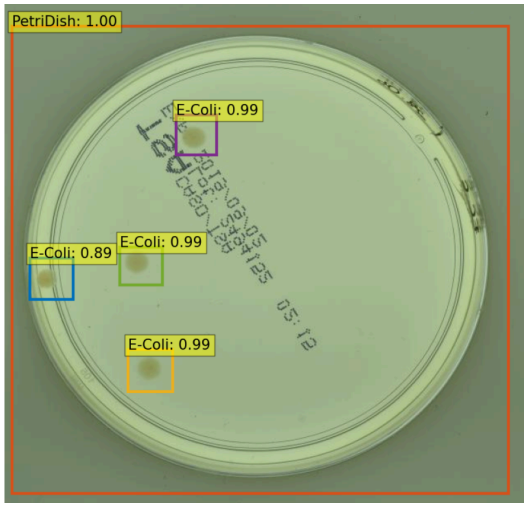
Фиг. 5. Визуализация на резултатите след прилагането на разработения модел

4.2. Оценката на модела е извършена с помощта на COCO метрики. Основните резултати показват средна точност (AP - Average Precision) от 0.375 при IoU(Intersection over Union)=0.50:0.95, среден остатък (AR - Average Recall) от 0.450 при IoU=0.50:0.95. Тези метрики демонстрират приложимостта на модела за големи масиви от данни, но демонстрират и слабости при прилагане върху малки обеми от данни.

5. Изводи и заключение

Настоящият проект демонстрира успешното приложение на трансформер - базирания модел DETR за задачата по идентифициране на бактериални колонии в лабораторно-снети изображения.

Средната точност (AP) от 0.375 и средния остатък (AR) от 0.450 показват, че моделът е способен да идентифицира и класифицира обекти с висока точност, особено когато става дума за големи масиви от данни, което е важно за скалируемостта на модела и неговото приложение. Въпреки това, малките набори от изображения представляват значително предизвикателство, което предполага нуждата от допълнителни подобрения в модела или в набора от данни. Интеграцията на COCO формат за анотации осигурява стандартен и добре документиран начин за обучение и оценка на модела. На Фигура 6 е представено примерно изображение с анотирани бактериални структури и увереността при тяхното класифициране.



Фиг. 6: Увереност в модела да разпознава бактерии от клас E-Coli.

Заключенията от проекта показват, че DETR е способен да се адаптира към специализирани задачи, като разпознаване на лабораторни проби, когато бъде обучен с подходящи данни и конфигурации. Бъдещата работа може да включва разширяване на набора от данни чрез включване на повече категории и изображения, както и използване на техники за ускоряване и повишаване скалируемостта на процеса на обучение, като semi-supervised learning или data augmentation.

6. Източници

Kaggle линк към notebook с резултати:
https://www.kaggle.com/code/jonicornpotato/big-data-tu

Github линк към notebook с резултати и документация:

https://github.com/joannaBlagoeva/tu_petri_4nika

[1] Makrai, L., Fodróczy, B., Nagy, S.Á. *et al.* Annotated dataset for deep-learning-based bacterial colony detection. *Sci Data* 10, 497 (2023). https://doi.org/10.1038/s41597-023-02404-8

[2] Pau, G., Fuchs, F., Sklyar, O., Boutros, M. & Huber, W. EBImage–an R package for image processing with applications to cellular phenotypes. Bioinformatics 26, 979–981, https://doi.org/10.1093/bioinformatics/btq046 (2010)

[3] Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. Nature Methods 9, 671–675, https://doi.org/10.1038/nmeth.2089 (2012)

[4] Geissmann, Q. OpenCFU, a new free and open-source software to count cell colonies and other circular objects. PloS ONE 8, e54072, https://doi.org/10.1371/journal.pone.0054072 (2013)

[5] Torelli, A. et al. AutoCellSeg: robust automatic colony forming unit (CFU)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques. Scientific Reports 8, 1–10, https://doi.org/10.1038/s41598-018-24916-9 (2018)

[6] Zhang, L. Machine learning for enumeration of cell colony forming units. Visual Computing for Industry. Biomedicine, and Art 5, 1–8, https://doi.org/10.1186/s42492-022-00122-3 (2022)

[7] Yang F, Zhong Y, Yang H, Wan Y, Hu Z, Peng S. Microbial Colony Detection Based on Deep Learning. *Applied Sciences*. 2023; 13(19):10568. https://doi.org/10.3390/app131910568

[8] Carion, Nicolas, et al. "End-to-end object detection with transformers." European conference on computer vision. Cham: Springer International Publishing, 2020 https://arxiv.org/pdf/2005.12872

[9] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016

[10] Hugging face Transformer библиотека - https://huggingface.co/docs/transformers/index

[11] Lighting AI Py-torch библиотека - https://lightning.ai/docs/pytorch/stable/

[12] MatPlotlib библиотека - Matplotlib documentation — Matplotlib 3.10.0 documentation