Санкт–Петербургский государственный университет

Калиев Темирлан Дауренович

Выпускная квалификационная работа

Методы распознавания изображений с люминофоров для вычисления параметров пучка в канале Бустер-Нуклотрон

Уровень образования: магистратура

Направление 02.04.02 «Фундаментальная информатика и информационные технологии»

Основная образовательная программа ВМ.5503

«Технологии баз данных»

Научный руководитель:
доцент, кафедра информационных и ядерных технологий, к.ф. - м.н., Сидорин Анатолий Олегович

Рецензент:
Главный инженер базовой установки Нуклотрон ЛФВЭ ОИЯИ, д.ф. - м.н., профессор Сыресин Евгений Михайлович

Санкт-Петербург
2022 г.

Оглавление

Введение 3

Постановка задачи 5

Обзор литературы 10

Глава 1. Методы распознавания образов 11

1.1. Two-Stage методы 11

1.2. One-Stage методы 12

Глава 2. Применение методов глубокого обучения 16

2.1. Подготовка данных 17

2.2. Tensorflow 18

2.3. PyTorch 23

Глава 3. Постобработка изображений и детектирование в реальном времени 25

3.1. Алгоритм предварительной обработки изображений 25

3.2. Система детектирования в реальном времени 26

Глава 4. Встройка системы детектирования в систему управления Tango Controls 28

Выводы 30

Заключение 31

Список литературы 32

Введение

В современном мире информационные технологии и искусственный интеллект все глубже проникают во все сферы жизни. Одной из крупных сфер в информационных технологиях является компьютерное зрение.

Компьютерное зрение – это область искусственного интеллекта (ИИ), которая стремится расширить возможности компьютеров по идентификации и определению объектов и людей на изображения и видео. Как и другие типы ИИ, компьютерное зрение ориентируется на выполнение и автоматизацию задач, имитирующих человеческие способности. Компьютерное зрение использует входные данные с сенсорных устройств, возможности ИИ, машинного обучения и глубокого обучения. В современных приложениях компьютерного зрения все чаще видна тенденция в сторону использования методов глубокого обучения. По сравнению с обычными (статистическими) методами глубокое обучение позволяет делать более точный анализ. Тем не менее статистические методы хорошо справляются с рядом определенных задач.

Сфер применения компьютерного зрения огромное количество. Среди них:

- Медицина [14]. Эта сфера характеризуется анализом изображений и постановки диагноза пациентам. Примером информации, полученной из такого рода изображений, является обнаружение опухолей, измерение органов, предоставление о строении мозга и т.д.
- Безопасность. Компьютерное зрение применимо для системы контроля доступа на основе распознавания лиц [18]: от офисов компаний до разблокировки смартфонов.
- Транспорт.

- Извлечение текста. Оптическое распознавание символов используется для обнаружения содержимого в данных с большим объемом текста, а также для автоматизации обработки текстовых документов.

В физике пучков заряженных частиц задача распознания образов и цифровой обработки изображений наиболее часто возникает при измерениях фазового объема пучка на выходе источников частиц и настройке каналов транспортировки пучков в случаях, когда чувствительным элементом датчика положения пучка является люминесцентный экран. Обычно форма пучка заранее известна с хорошей точностью, поэтому задача оцифровки изображения может быть решена и без применения методов компьютерного зрения. Ситуация меняется в случае импульсного режима работы источника частиц или ускорителя, когда информацию необходимо получить в промежутке между импульсами. Благодаря высокому быстродействию при незначительной потери точности алгоритмы распознавания образов становятся конкурентоспособными. Кроме того, при разработке конкретного приложения можно использовать достаточно отработанные на других задачах библиотеки, что существенно сокращает и срок создания готового продукта.

Постановка задачи

Задача обнаружения объектов одна из наиболее популярных в компьютерном зрении. Применение этой задачи позволило оптимизировать процессы в многих сферах жизни. Одним из примеров является автоматизация процессов на ускорительной установке NICA в Объединенном институте ядерных исследований.

Объединённый институт ядерных исследований (ОИЯИ) — крупный международный научный проект, учреждённый восемнадцатью странами-участницами в 1956 году. Цель его создания – экспериментальные и теоретические исследования в области ядерной физики, физики элементарных частиц и физики конденсированного состояния. На базе ОИЯИ в 2013 году было запущено строительство ускорительного комплекса NICA (Nuclotron based Ion Collider fAcility). Проект NICA нацелен на воспроизведение и изучение свойств ядерной материи в условиях, соответствующих ранним стадиям развития Вселенной (кварк-глюонная плазма). Ускорение тяжелых ионов производится цепочкой ускорителей, включающей в себя линейный ускоритель тяжелых ионов и два сверхпроводящих циклических ускорителей – синхротронов: Бустера и основного ускорителя комплекса – Нуклотрона. Нуклотрон был введен в эксплуатацию в 1993 году, и с тех пор использовался для проведения исследований. Для повышения интенсивности и энергии ускоренных пучков Нуклотрона в 2020 году был запущен промежуточный ускоритель – Бустер, а в 2021 году были смонтированы система быстрого вывода пучка из Бустера и канал транспортировки пучка из Бустера в Нуклотрон. Ускорители имеют разные радиусы повороту пучка и расположены на разной высоте, поэтому канал имеет сложную трехмерную геометрию.

Схема компоновки канала транспортировки пучка Бустер-Нуклотрон приведена на Рис. 1. Фрагмент модели участка быстрого вывода пучка из

Бустера и канала транспортировки пучка Бустер-Нуклотрон приведен на Рис. 2.

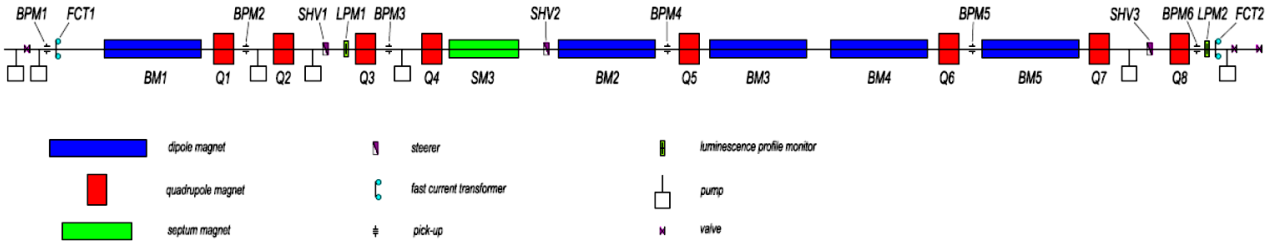

Рис.1. Компоновка канала транспортировки пучка Бустер-Нуклотрон.

Поворотные магниты BM1 – BM5 формируют траекторию пучка, а квадрупольные магниты Q1 – Q8 осуществляют его фокусировку и согласование в точке инжекции в Нуклотрон. Септумный магнит SM3 предназначен для выделения из пучка ионов в требуемом зарядовом состоянии. Положение элементов магнитной системы в реальном канале отличается от идеального из-за погрешностей юстировки, поэтому необходимо проводить исследование и настройку канала. С этой целью в канале размещены устройства диагностики пучка и коррекции орбиты. Для измерения интенсивности пучка на входе и выходе канала используются быстрые трансформаторы тока, положение центра тяжести пучка измеряется двумя координатными мониторами положения пучка BPM1 – BPM6, представляющими собой емкостные датчики или пикап электроды. Коррекция орбиты пучка осуществляется с помощью двух координатных корректирующих дипольных магнитов (steerer) SHV1 – SHV3. Согласование пучка со входом в Нуклотрон осуществляется за счет подстройки токов питания квадрупольных магнитов. При исследовании канала измеряется матрица отклика – изменение положения пучка при малых изменениях токов квадрупольных и корректирующих магнитов. Дополнительную информацию о форме и положении пучка можно получить с помощью люминесцентных

датчиков – люминофоров LPM1, LPM2, которые могут дистанционно вводится в канал.

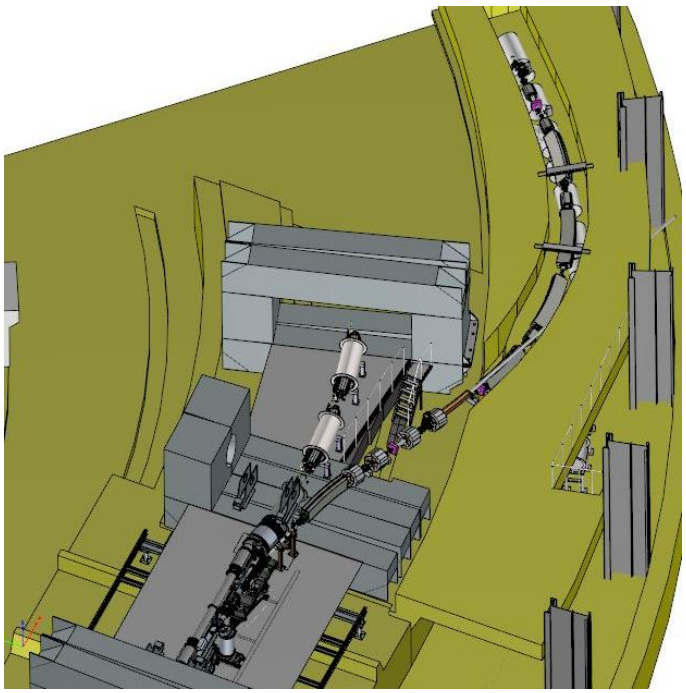

Рис. 2. Внешний вид участка быстрого вывода пучка из Бустера и канала транспортировки пучка Бустер-Нуклотрон (фрагмент модели).

Пучок, попадая на люминофор, вызывает его свечение, и изображение пучка с помощью CCD камеры передается в систему управления. Для построения матрицы отклика необходимо по этому изображению определить такие количественные характеристики, как: положение центра тяжести, отношение полуосей представляющего эллипса, угол наклона большой полуоси, или их производные. Первоначально программа должна зафиксировать, что пучок прошел по каналу, а потом определить форму этого пучка. В общем случае пучок имеет форму эллипса (Рис. 4), в частном случае – окружности, и при прохождении пучка далеко от расчетной траектории он может попадать на край люминофора, при этом он имеет неправильную форму (Рис. 3).

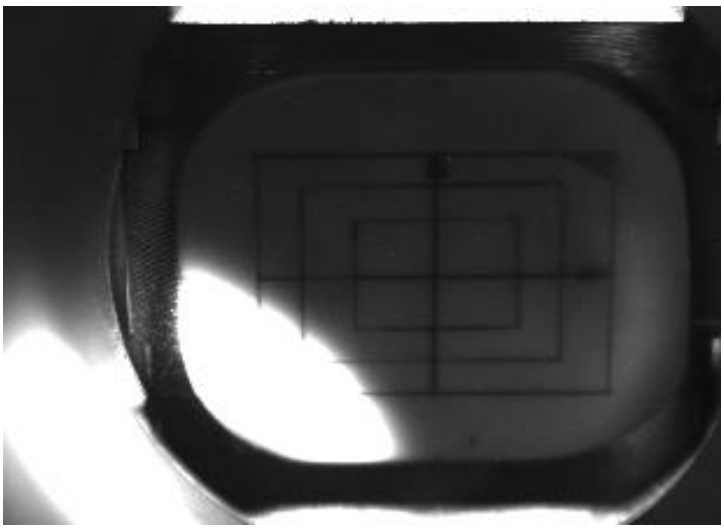

Рис. 3. Пример формы неудачного пучка частиц

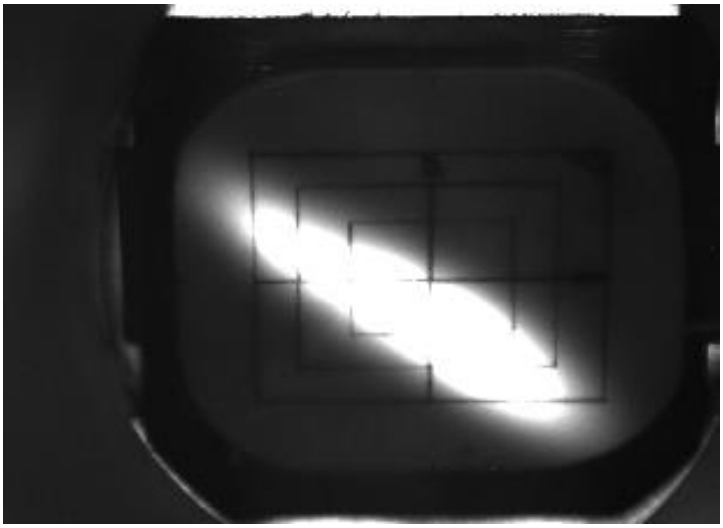

Рис. 4. Пример пучка частиц в форме эллипса

В данной работе для получения количественных характеристик изображения пучка использовалась система компьютерного зрения. Для ее реализации были поставлены следующие задачи:

- Изучить методы глубокого обучения, в том числе методы распознавания изображений;
- Осуществить выбор наиболее оптимальных методов на основе анализа объекта управления;

- Получить изображения пучка в ходе сеанса ускорительного комплекса в достаточном количестве для обучения моделей;
- Изучить инструменты трансферного обучения и провести обучение двух наиболее оптимальных моделей;
- В необходимом объеме изучить систему управления ускорительным комплексом;
- Создать класс устройства с необходимыми параметрами.

Обзор литературы

Применений технологий компьютерного зрения огромное количество. Примеры таких применений приведены в [11, 14, 18].

В ходе исследования были изучены различные сферы компьютерного зрения. Детектирование объектов в реальном времени подробно описано в [5]. Примеры применения сегментации изображений описаны в [6, 11]. Описание архитектуры сверточной нейронной сети и ее место в компьютерном зрении приведено в [18, 19].

Описание терминов и более подробная работа моделей SSD изложена в [16], Yolov в [5]. Для обучения моделей глубокого обучения использовались фреймворки Tensorflow, PyTorch. Подробную информацию об этих инструментах можно прочитать в официальной документации, Tensorflow – [22], PyTorch – [21].

Аналогов исследований применения методов компьютерного зрения для распознавания пучков частиц с люминофоров не существует. Данная задача похожа на задачу обнаружения объектов в сфере медицины, а также сегментации медицинских снимков. Применение методов компьютерного зрения в медицине описано в [14].

Глава 1. Методы распознавания образов

Среди методов глубокого обучения существуют огромное количество архитектур моделей для распознавания образов. В моделировании компьютерного зрения доминируют сверточные нейронные сети (CNN). Начиная с AlexNet [1] и ее революционной производительности в задаче классификации изображений ImageNet. Архитектура CNN с тех пор развивалась и становилась более мощной благодаря большему масштабу [6, 10], сложным формам свертки [9, 4, 17] и обширным соединениям [2].

CNN являются стандартными моделями нейронных сетей в компьютерном зрении. С усложнением архитектуры появились такие модели, как VGG, GoogleNet, ResNet, DenceNet [9], EfficientNet и другие. В задаче детектирования объектов существует множество подходов для решения. Наиболее применимыми являются One-Stage и Two-Stage подходы.

1.1. Two-Stage подходы

В Two-Stage подходах используются 2 компонента, Region-based Convolutional Neural Network (RCNN) для предсказаний «регионов интереса» (RoI's), внутри которых могут быть объекты, и уже эти RoI классифицирует и уточняет координаты боксов CNN.

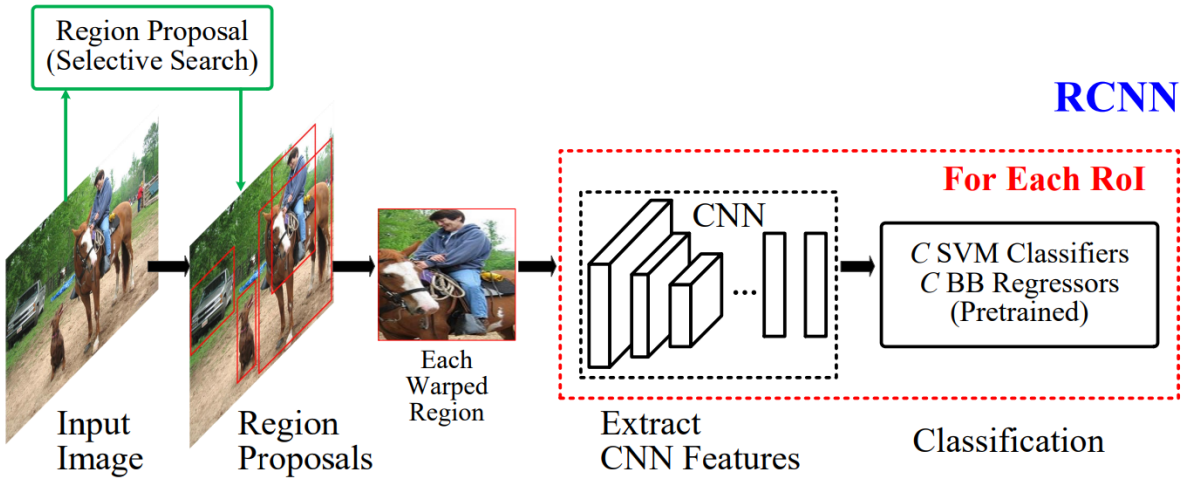

Рис 5. Схема RCNN

Впоследствии появились улучшения RCNN, такие как Fast-RCNN [5], Faster-RCNN и Mask-RCNN.

1.2. One-Stage подходы

Семейство алгоритмов R-CNN обеспечивает высокую точность благодаря сложности своей архитектуры, однако из-за этого такие алгоритмы могут быть медленными и не применимыми для задач, где необходимо распознавание объектов в реальном времени. One-Stage подходы, в отличие от Two-Stage, не используют дополнительный алгоритм для генерации боксов. One-Stage получили широкое применение за счет малого времени ожидания и высокой производительности по различным метрикам оценки.

One-Stage моделям требуется только один проход изображения через нейронную сеть, после чего они предсказывают все боксы за один раз. К таким моделям относятся YOLO (You only look once) [5], SSD (Single shot detector), DetectNet и другие. Логика таких моделей схожа: на вход подается изображение, а на выходе – 5 чисел, 4 из которых это координаты бокса и 1 – уровень уверенности.

YOLOv5 – одна из самых быстрых и точных моделей, используемых для обнаружения объектов в реальном времени. Данная модель, разработанная Ultralytics, была выпущена в июне 2020 года [15]. Среди моделей YOLOv5 существует классификация: n, s, m, l, x. Каждая их них обладает разной точностью и производительностью (Рис. 6).

Главные различия YOLOv5 от предыдущих версий заключаются в том, что для основы (Backbone) YOLOv5 используется CSPdarknet со структурой Focus (Рис. 7). Преимущество использования слоя Focus заключается в уменьшенном объеме требуемой памяти CUDA, уменьшении количества слоев, увеличении прямого и обратного распространения [3].

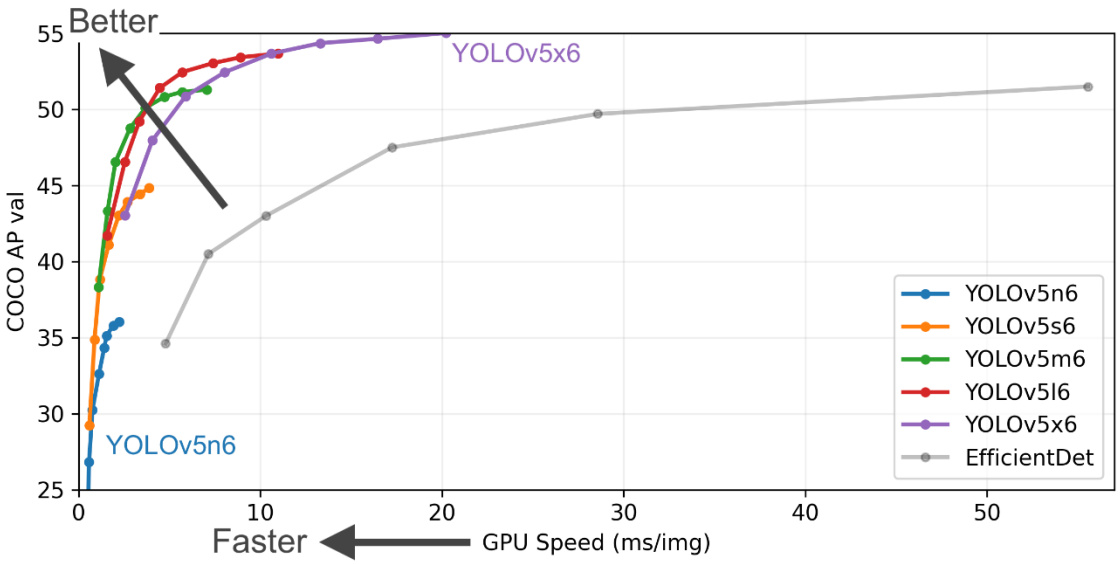

Рис. 6. Сравнение моделей YOLOv5 [15]

На рис. 6 изображены графики производительности моделей Yolov5. На горизонтальной оси определено время обучения моделей. Чем оно меньше, тем быстрее модель обучается. По вертикальной оси определена средняя точность модели (average precision) на датасете COCO. Самой быстрой моделью является Yolov5n6, однако имеет самую низкую точность по сравнению с другими моделями. Наибольшей точностью обладает модель Yolov5x6, но для ее обучения потребуется большее время.

Архитектура Yolov5 состоит из трех основных блоков: Backbone, Neck, Head. Backbone в данном случае является сверточная нейронная сеть CSPDarknet. CSPDarknet уменьшает параметры и время операций с плавающей запятой в секунду (FLOPS). Это позволяет повысить скорость модели и уменьшить ее размер. Neck – PANet (Path Aggregation Network), модель, которая позволяет увеличить поток информации. PANet улучшает использование точных сигналов локализации на нижних уровнях, что может повысить точность определения местоположения объекта. Head – слой Yolov, генерирующий карты признаков трех размеров (18 × 18, 36 × 36, 72 × 72), для достижения многомасштабного [7] предсказания, что позволяет модели обрабатывать небольшие, средние и большие объекты.

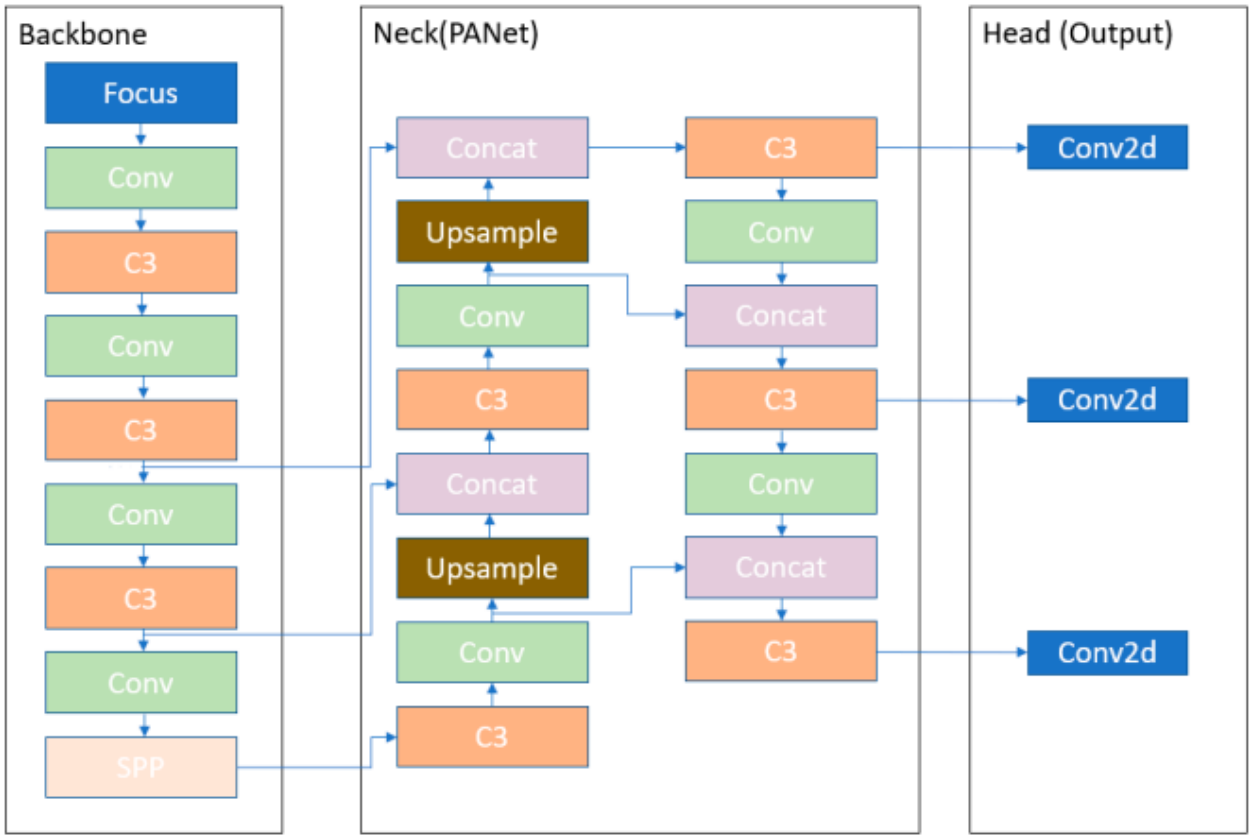

Рис. 7. Архитектура модели YOLOv5l

SSD модели на ряду с YOLO также относятся к One-stage методам. Они состоят из Backbone и выхода (Head). Backbone модель обычно предварительно обученная сеть, предназначенная для извлечения признаков. Head – это один или несколько сверточных слоев. Архитектура SSD модели представлена на Рис. 8, первые слои (белые фигуры) являются частью Backbone, последние слои (синие фигуры) – Head.

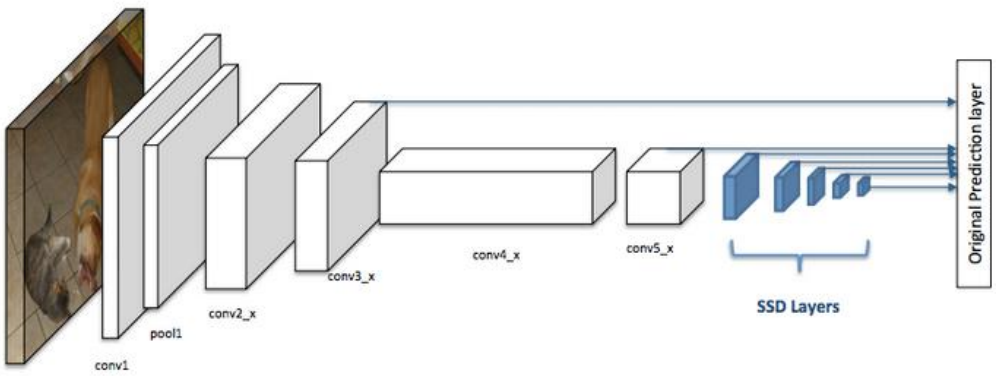

Рис. 8. Архитектура SSD модели

В зависимости от того, какая модель используется в качестве Backbone, существует множество вариаций данной архитектуры. Данный вариант архитектуры дает баланс между точностью, которую демонстрируют лучше всего Two-Stage модели, и скоростью, в которой YOLO показывает лучшие результаты.

Глава 2. Применение методов глубокого обучения

Методы глубокого обучения достигли огромных успехов в фундаментальных задачах компьютерного зрения, таких как распознавание изображений [6], обнаружение объектов [5, 8, 12, 16] и сегментация изображений [6, 11].

Существует множество предварительно обученных моделей. На ускорительном комплексе NICA наиболее перспективным является применение методов машинного обучения при исследовании и настройке канала перевода пучка из Бустера в Нуклотрон. Предварительная настройка канала заключается в задании токов питания поворотных и квадрупольных магнитов, рассчитанных по идеальной модели канала. После этого, за счет небольших вариаций этих токов обеспечивается достаточная для дальнейшей настройки эффективность прохождения пучка по каналу. Далее измеряется матрица отклика пучка на воздействие корректирующих магнитов и небольших вариаций тока квадрупольных магнитов. Решением задачи оптимизации восстанавливаются реальные положения магнитов канала и связь между током и магнитным полем для каждого конкретного магнита. Эту задачу необходимо решать в начале работы с каждым новым сортом ионов и с новым значением их кинетической энергии, поэтому автоматизация процедуры построения матрицы отклика обеспечивает значительную экономию дорогостоящего пучкового времени. При этом существенную роль играет быстродействие применяемого алгоритма распознавания образов. Сгустки заряженных частиц проходят по каналу с интервалом 4 – 5 секунд, и за это время система управления должна получить количественные данные от всех датчиков и перестроить ток одного из элементов коррекции. Для таких целей подходят one-stage модели, например, YOLOV5, SSD и другие.

Исходя из сравнения производительности моделей на Рис. 9, были выбраны модели YOLOV5 и SSD MobileNet. Модели YOLOV5 использовалась

с помощью библиотеки PyTorch [21], SSD MobileNet с помощью TensorFlow [13, 22].

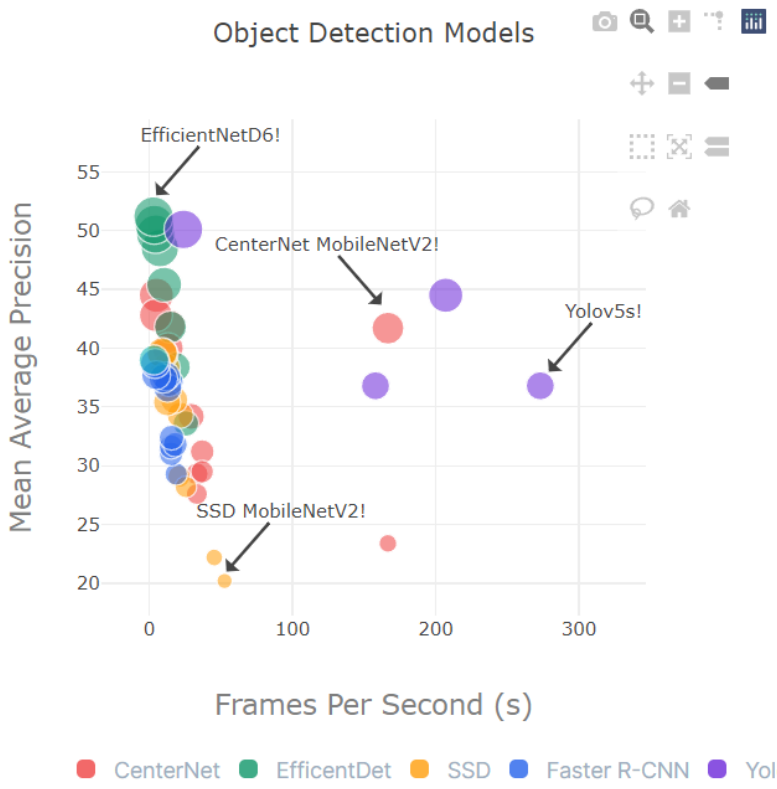

Рис. 9. Сравнение производительности различных моделей [13, 15]

2.1. Подготовка данных

В ходе сеанса ускорительного комплекса в феврале – марте 2022 года были получены 60 изображений пучка, которые составили датасет для обучения моделей. Все изображения были размечены с помощью пакета LabelImg. LabelImg – это бесплатный инструмент с открытым кодом для графической маркировки изображений. Он поддерживает разные форматы маркировки (PasvalVOC, Yolo, CreateML). Часть изображения, на которой присутствует пучок частиц маркируется таким образом, как на Рис. 10. После маркировки создается файл, в котором содержатся все детали изображения, включая координаты объекта, тип объекта и др. Датасет был поделен на

тренировочную и тестовую выборки в соотношении 50 и 10 изображений соответственно. Количество классов – 3 (ellipse, circle, bad_beam).

Рис. 10. Маркировка выбранной области

2.2. Tensorflow object detection

Предложенная система предназначена для детектирования пучков частиц и определение их формы в реальном времени. Первый вариант с моделью SSD MobileNet использует Tensorflow object detection (TFOD) API 2.0 и дообучается с помощью трансферного обучения на тренировочном датасете.

Для обучения с помощью TFOD создаются записи для тренировочной и тестовой выборки в формате TF record. TF record – бинарный формат записи данных для Tensorflow. Такой формат записи значительно ускоряет процесс обучения и занимает меньше места.

TFOD API, являющийся фреймворком с открытым кодом, упрощает разработку, обучения и развертывание моделей для детектирования объектов. Внутри TFOD существует отдельный инструмент Tensorflow object detection model zoo [13], который предоставляет огромный выбор моделей (Рис. 11).

Среди моделей, относящихся к SSD MobileNet, была выбрана SSD MobileNet v2 FPNlite 320x320 [13]. FPNlite (Feature Pyramid Network) – это экстрактор признаков, принимающий на вход одномасштабное изображение произвольного размера и выводящий карту признаков. Данная модель предварительно обучена на изображениях из датасета COCO 2017, масштабированными до размера 320x320. Впоследствии модель была дообучена на 50 размеченных изображениях с люминофорами. В ходе обучения была применена аугментация данных путем случайной обрезки, случайного вертикального и горизонтального поворота, изменения контраста. Модель была обучена на 10000 шагах. Результаты предсказания модели SSD MobileNet v2 FPNlite 320x320 на тестовом датасете из 10 изображений приведены в таблице 1.

AP (Average Precision)	AR (Average Recall)
0.85	0.867

Таблица 1. Результаты предсказаний SSD MobileNet

TensorFlow 2 Detection Model Zoo

TensorFlow 2.2 Python 3.6

We provide a collection of detection models pre-trained on the COCO 2017 dataset. These models can be useful for out-of-the-box inference if you are interested in categories already in those datasets. You can try it in our inference colab

They are also useful for initializing your models when training on novel datasets. You can try this out on our few-shot training colab.

Please look at this guide for mobile inference.

Finally, if you would like to train these models from scratch, you can find the model configs in this directory (also in the linked `tar.gz`s).

Model name	Speed (ms)	COCO mAP	Outputs
CenterNet HourGlass104 512x512	70	41.9	Boxes
CenterNet HourGlass104 Keypoints 512x512	76	40.0/61.4	Boxes/Keypoints
CenterNet HourGlass104 1024x1024	197	44.5	Boxes
CenterNet HourGlass104 Keypoints 1024x1024	211	42.8/64.5	Boxes/Keypoints
CenterNet Resnet50 V1 FPN 512x512	27	31.2	Boxes
CenterNet Resnet50 V1 FPN Keypoints 512x512	30	29.3/50.7	Boxes/Keypoints
CenterNet Resnet101 V1 FPN 512x512	34	34.2	Boxes
CenterNet Resnet50 V2 512x512	27	29.5	Boxes
CenterNet Resnet50 V2 Keypoints 512x512	30	27.6/48.2	Boxes/Keypoints
CenterNet MobileNetV2 FPN 512x512	6	23.4	Boxes
CenterNet MobileNetV2 FPN Keypoints 512x512	6	41.7	Keypoints
EfficientDet D0 512x512	39	33.6	Boxes
EfficientDet D1 640x640	54	38.4	Boxes
EfficientDet D2 768x768	67	41.8	Boxes
EfficientDet D3 896x896	95	45.4	Boxes
EfficientDet D4 1024x1024	133	48.5	Boxes
EfficientDet D5 1280x1280	222	49.7	Boxes
EfficientDet D6 1280x1280	268	50.5	Boxes
EfficientDet D7 1536x1536	325	51.2	Boxes
SSD MobileNet v2 320x320	19	20.2	Boxes
SSD MobileNet V1 FPN 640x640	48	29.1	Boxes
SSD MobileNet V2 FPNLite 320x320	22	22.2	Boxes
SSD MobileNet V2 FPNLite 640x640	39	28.2	Boxes

Рис. 11. Tensorflow object detection model zoo

В ходе обучения у модели есть несколько функций потери такие как, потеря классификации (classification loss), потеря локализации (localization loss) и потеря регуляризации (regularization loss).

Потеря классификации (L_{cls}) определена как потеря softmax (softmax_cross_entropy_with_logits в Tensorflow) по нескольким классам. Потеря классификации вычисляется по фор. (1).

$$\boldsymbol{L_{cls}} = -\sum_{i\in pos} \mathbf{1}_{ij}^{k} \log\left(\widehat{c_i^k}\right) - \sum_{i\in neg} \log\left(\widehat{c_i^0}\right) \quad (1)$$

где $\widehat{c_i^p} = \exp(\widehat{c_i^p})/\sum_p \exp(\widehat{c_i^p}) = softmax(\widehat{c_i^p})$. 1_{ij}^{k} указывает совпадает ли i-я ограничительная рамка и j-я истинная рамка для объекта в классе k.

Потеря локализации предстваляет собой сглаженную (smooth) функцию L1 между коррекцией ограничительной рамки и истинной рамкой. Формула потери локализации представлена в (2) – (6).

$$\boldsymbol{L_{loc}(x, l, g)} = \sum_{i\in pos}^{N} \sum_{m\in\{cx,cy,w,h\}} \boldsymbol{x_{ij}^{k}\, smooth_{L1}(l_i^m - \widehat{g_j^m})} \quad (2)$$

$$\boldsymbol{\widehat{g_j^{cx}} = (g_j^{cx} - d_i^{cx})/d_i^w} \quad (3)$$

$$\boldsymbol{\widehat{g_j^{cy}} = (g_j^{cy} - d_i^{cy})/d_i^h} \quad (4)$$

$$\boldsymbol{\widehat{g_j^w} = \log\left(\frac{g_j^w}{d_i^w}\right)} \quad (5)$$

$$\boldsymbol{\widehat{g_j^h} = \log\left(\frac{g_j^h}{d_i^h}\right)} \quad (6)$$

где, N – количество совпадающих рамок, l – предсказываемая ограничительная рамка, g – истинная ограничивающая рамка, $\hat{g}$ – закодированная истинная ограничивающая рамка, x_{ij}^{k} – индикатор соответствия между предсказываемой ограничительной рамкой i и истинной j категории k.

На рис. 12, 13, 14 изображены функции потерь для модели SSD MobileNet v2 FPNlite с помощью интрумента TensorBoard. На каждом графике по горизонтальной оси определены шаги обучения.

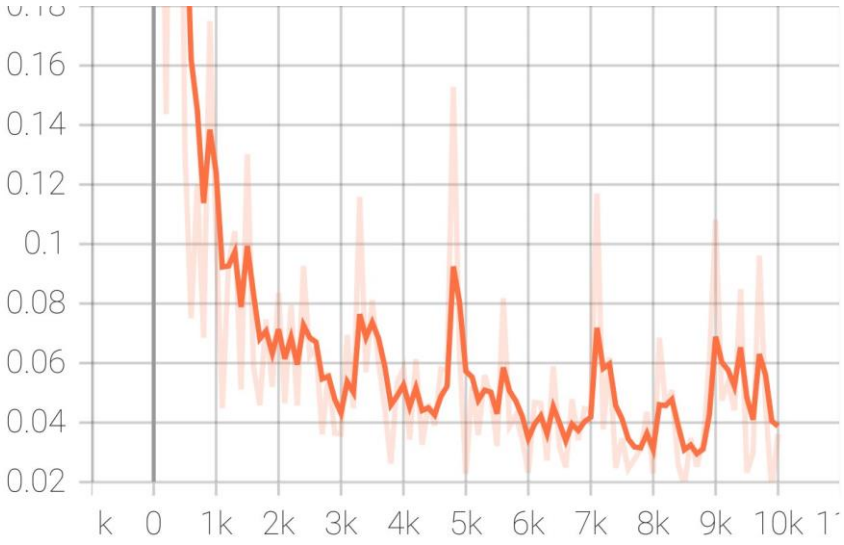

Рис. 12. Потеря классификации в ходе обучения SSD MobileNet v2

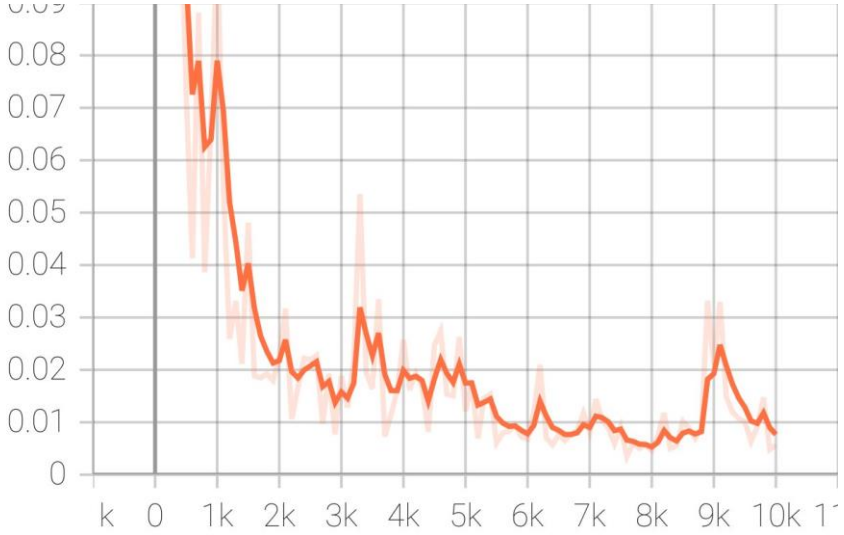

Рис. 13. Потеря локализации в ходе обучения SSD MobileNet v2

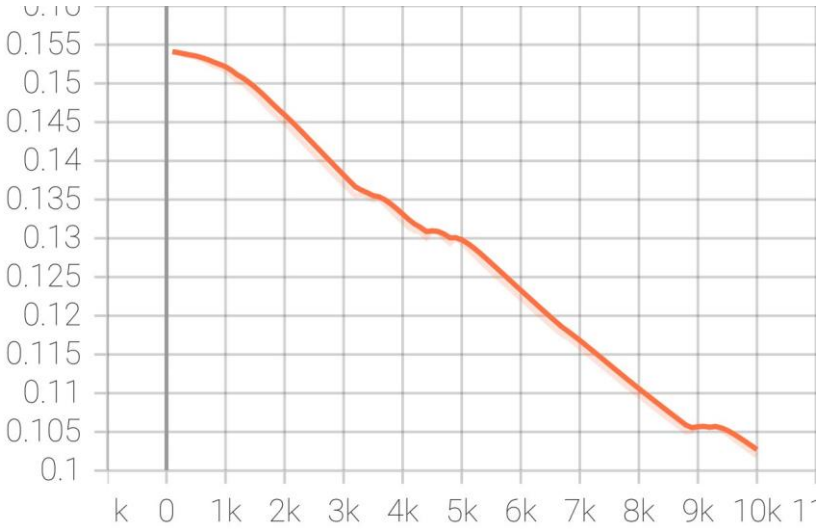

Рис. 14. Потеря регуляризации в ходе обучения SSD MobileNet v2

2.3. PyTorch

PyTorch – это библиотека глубокого обучения. Также, как и Tensorflow является популярным фреймворком для глубокого обучения, компьютерного зрения и др. В отличие от Tensorflow, в PyTorch вычисление графов происходит динамически. Такой подход дает большие возможности, позволяет использовать все возможности языка программирования для вычислений.

Как и Tensorflow, PyTorch предоставляет множество моделей. Семейство Yolov5 работает под управлением PyTorch. Из всех моделей Yolov5 для обучения была выбрана самая модель Yolov5x6. Данная модель была предварительно обучена на изображениях из датасета COCO.

Модель была обучена на 40 эпохах. Результаты предсказания модели Yolov5x6 на тестовом датасете приведены в таблице 2:

AP (Average Precision)	AR (Average Recall)
0.912	1.0

Таблица 2. Результаты предсказаний модели Yolov5x6

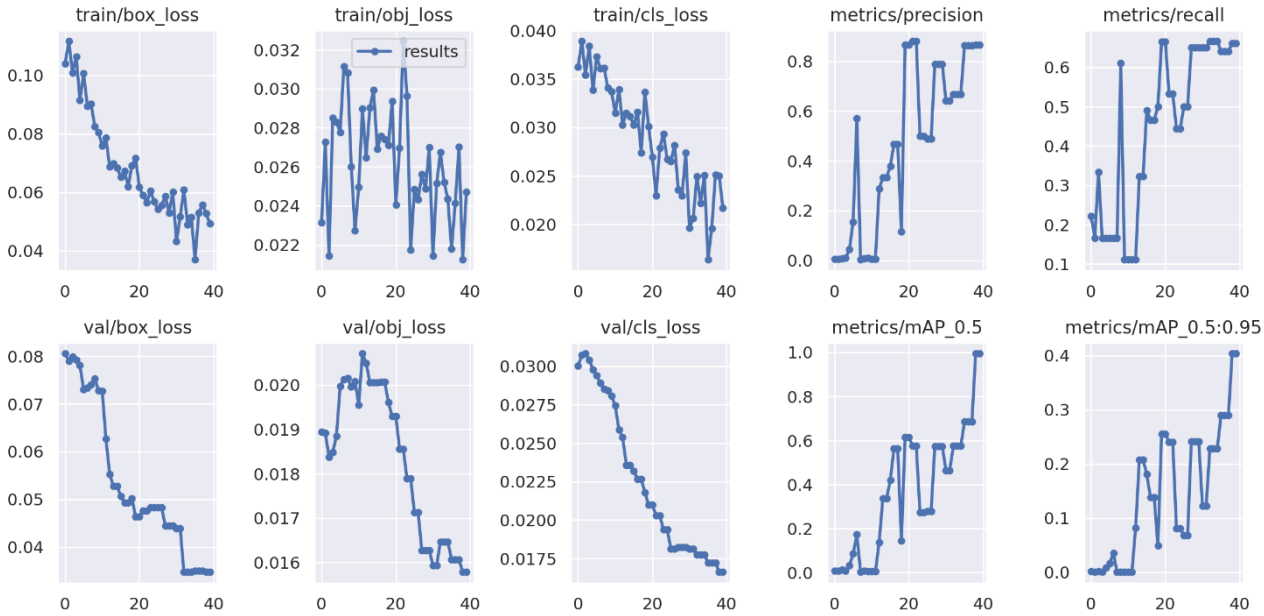

Рис. 15. Дополнительные результаты обучения Yolov5x6

На рис. 15 изображены функции потерь в ходе обучения модели Yolov5x6, а также метрики Precision и Recall. По горизонтальной оси отложены эпохи обучения.

По итогам обучения модель Yolov5x6 показала результат лучше по сравнению с SSD MobileNet v2 FPNlite.

Глава 3. Постобработка изображений и детектирование в реальном времени

Помимо детектирования пучков частиц и распознавание их формы требуется последующее вычисление параметров формы пучка, например, координаты центра масс, смещение центра масс относительно центра сетки, площадь, эксцентриситет и другие. Процедура обработки изображений, вычисление параметров пучка частиц осуществляется с помощью библиотеки компьютерного зрения OpenCV [14].

OpenCV – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. OpenCV предоставляет функции, которые в первую очередь нацелены на компьютерное зрения в реальном времени. Данная библиотека является основой для многих приложений на основе компьютерного зрения. OpenCV содержит более 2500 эффективных алгоритмов компьютерного зрения и машинного обучения, которые используются для обнаружения и распознавания лиц, идентификации объектов, отслеживания движения объектов и многого другого.

3.1. Алгоритм предварительной обработки изображений

Процесс предварительной обработки изображений состоит из:

- Применение размытия по Гауссу;
- Перевод изображения из цветового пространства RGB в HSV;
- Обнаружение границ с помощью оператора Кэнни;
- Морфологические преобразования (эрозия и расширение).
- Нахождение контура и подсчет параметров формы пучка частиц.

На этапе применения размытия по Гауссу применяется функция cv2.GaussianBlur из библиотеки OpenCV. Размытие по Гауссу позволяет добиться улучшения структуры изображения. Далее происходит преобразование изображения из цветового пространства RGB в пространство HSV (Hue, Saturation, Value – тон, насыщенность, значение). Оператор может регулировать все параметры пространства HSV с помощью маски для выделения пучка частиц из остальной части изображения. После этого происходит обнаружение границ пучка частиц с помощью оператора Кэнни. Следующим этапом являются морфологические преобразования, а именно – эрозия и расширение. На получившемся изображении происходит финальная процедура – нахождение контура и подсчет параметров формы пучка частиц. Считаются такие параметры, как: длина большой и малой полуосей, фокальное расстояние, площадь, длина дуги эллипса, эксцентриситет и расстояние от центра сетки до центра масс пучка.

3.2. Система детектирования в реальном времени

Система детектирования реализована с помощью OpenCV и видеокамеры. Система детектирует пучок частиц, определяет его форму, проделывает обработку изображения и подсчитывает все нужные параметры как на рис. 16. В верхнем левом углу изображен результат алгоритма обработки изображения. В левом нижнем углу изображены предсказание формы пучка и его границы. Значение в процентах означает уровень достоверности предсказания. В верхнем правом углу находится исходное изображение. В нижнем правом отображаются параметры формы пучка.

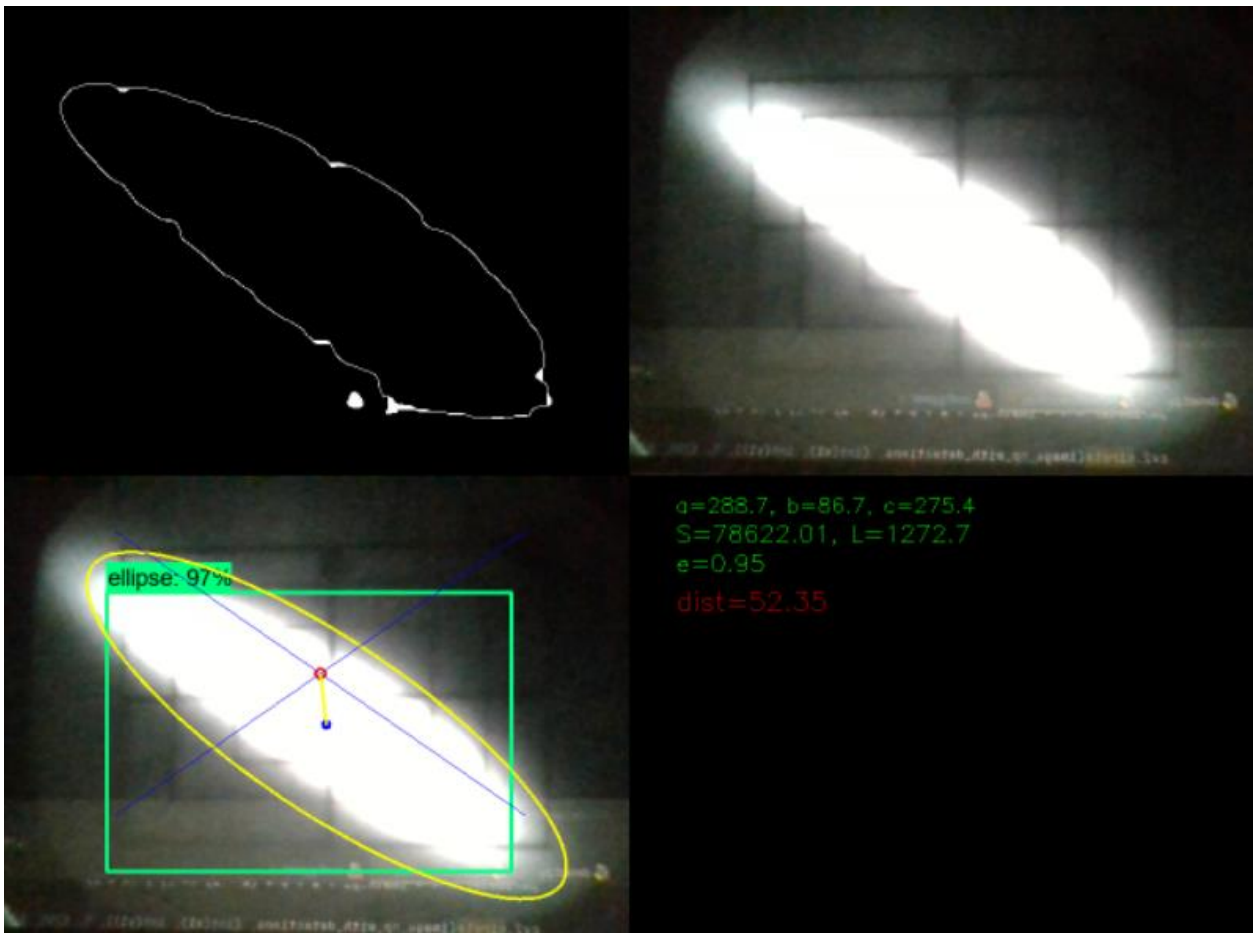

Рис. 16. Пример обработки изображения для вычисления параметров формы пучка и детектирование с помощью модели глубокого обучения

Глава 4. Встройка системы детектирования в систему управления Tango Controls

Система управления Tango Controls – это набор инструментов с открытым исходным кодом, позволяющий управлять любым оборудованием или программным обеспечением. Данная система основана на принципе распределенных устройств. Устройство Tango Controls представляет собой отражение реального устройства, либо логической сущности. Система Tango Controls поддерживает такие языки программирования, как Python, Java, C++.

Каждое устройство Tango Controls предназначено для определенной задачи. Устройства обмениваются между собой сообщениями со стандартным интерфейсом. У каждого устройства есть свои атрибуты (Attributes). Атрибутами называются поля данных с определенным типом (логический, строковый, целое число, число с плавающей точкой, символы, перечисления и др.) Атрибуты могут иметь разную размерность: скалярные (Scalar Attributes), спектральные (Spectrum Attributes) – одномерные массивы и изображения (Image Attributes) – двумерные массивы.

Система детектирования пучков частиц записывает параметры формы пучка и другие в виде атрибутов в специально разработанное Tango устройство CollectorDS. Данное устройство представляет атрибуты с других устройств в виде своих атрибутов. Основные предназначения CollectorDS:

- Агрегирование данных от нескольких устройств для минимизации сетевого трафика. Запросы от клиентов идут только к коллектору, а он сам решает, когда нужно обновлять данные у устройств-источников;
- Взаимодействие между разными установками Tango Controls. В качестве источников можно настроить атрибуты и команды устройств, находящихся в другом TANGO_HOSTe, четко оформив таким образом правила взаимодействия между хостами.

- Создание устройства с простыми атрибутами, не связанными с каким-либо источником данных. В этом случае данные могут быть записаны сторонними клиентскими приложениями через Tango или веб-страницами по протоколу REST.

Обновление атрибутов происходит каждые 5 секунд в ходе работы программы. Tango устройство имеет семь скалярных атрибутов и один атрибут типа изображение. Атрибут изображения является черно-белым изображением из системы детектирования в реальном времени как на рис. 14.

Скалярные атрибуты:

- MeasStatus – результат измерения пучка, 1 – хороший пучок (ellipse, circle), 0 – плохой пучок (bad_beam);
- Timestamp – временная метка, показывает время последнего измерения;
- Eccentricity – эксцентриситет;
- Square – площадь;
- X, Y – координаты центра масс пучка;
- Distance – расстояние от центра сетки до центра масс пучка.

Запись атрибутов в Tango устройство можно продемонстрировать следующим псевдокодом:

```
IF beam in (“ellipse”, “circle”)
    THEN MeasStatus =1
    Device.write_all_attributes()
ELSE
    MeasStatus = 0
    Device.write_all_attributes()
```

Выводы

В ходе выполнения работы были изучены методы распознавания изображений. Для этого были изучены методы глубокого обучения.

На основе анализа требований, предъявляемых процедурой исследования и настройки канала транспортировки пучка из Бустера в Нуклотрон, для обучения выбраны две модели Yolov5x6 и SSD MobileNet v2 FPNlite 320x320.

Для обучения моделей в ходе сеанса ускорительного комплекса получены 60 изображений пучка. Датасет был поделен на тренировочную и тестовую выборки в соотношении 50 и 10 изображений соответственно.

Были изучены инструменты Tensorflow и PyTorch. С помощью трансферного обучения эти модели были дообучены на изображениях с люминофоров.

Были изучена система управления Tango Controls, используемая на ускорительном комплексе NICA. Написан класс устройства с 6-ю скалярными атрибутами, передающими количественные параметры формы изображения пучка.

Заключение

В результате выполнения данной работы была разработана система компьютерного зрения, способная по изображению пучка частиц на люминофоре определять его форму и считать геометрические параметры формы, а также передавать эти параметры в систему управления Tango Controls. Было достоверно продемонстрировано, что обученные модели обеспечивают быстродействие и разрешение, необходимые для автоматизации процедуры настройки канала перевода пучка из Бустера в Нуклотрон комплекса NICA. Тестирование отдельных элементов системы, набор данных для обучения были проведены в ходе работы ускорительного комплекса в феврале-марте 2022 года. Внедрение разработанной системы запланировано на ближайший сеанс работы, проведение которого намечается в сентябре – октябре 2022 года.

Исходный код разработанного программного обеспечения доступен в GitHub по ссылке [26].

Список литературы

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, с. 1097–1105, 2012

2. Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, с. 4700–4708, 2017

3. Glenn-jocher, YOLOv5 Focus() Layer #3181. In Ultralytics: Github; 2021 [Электронный ресурс] URL: https://github.com/ultralytics/yolov5/discussions/3181m1 (дата обращения 24.02.2022)

4. Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, с. 764–773, 2017

5. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 779–788.

6. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 770–778

7. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv: 1804.02767.

8. Ross Girshick. 2015. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision. 1440–1448.

9. Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition, с. 1492–1500, 2017.

10. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In BMVC, 2016.

11. Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. 2018. Textsnake: A flexible representation for detecting text of arbitrary shapes. In European Conference on Computer Vision. 20–36.

12. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems. 91–99.

13. Tensorflow 2.0 Object Detection API model zoo [Электронный ресурс] URL: https://github.com/tensorflow/models/blob/master/research/object_detection/ (дата обращения 18.04.2022)

14. Topol, E. J. High-performance medicine: the convergence of human and artificial intelligence. Nat. Med. 25, 44–56 (2019).

15. Ultralytics Yolov5 Benchmarks. [Электронный ресурс] URL: https://github.com/ultralytics/yolov5/ (дата обращения 15.04.2022)

16. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In European Conference on Computer Vision. Springer, 21–37.

17. Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, с. 9308–9316, 2019.

18. Xuepeng Shi, Shiguang Shan, Meina Kan, Shuzhe Wu, and Xilin Chen. 2018. Realtime rotation-invariant face detection with progressive calibration networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2295–2303

19. Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv preprint: 2103.14030, 2021

20. Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. Fcos: Fully convolutional one-stage object detection. In IEEE International Conference on Computer Vision. 9627–9636.

21. Документация PyTorch [Электронный ресурс] URL: https://pytorch.org/docs/stable/index.html (дата обращения 12.05.2022)

22. Документация Tensorflow [Электронный ресурс] URL: https://www.tensorflow.org/api_docs (дата обращения 12.05.2022)

23. Официальный сайт OpenCV [Электронный ресурс] URL: https://opencv.org/ (дата обращения 17.05.2022)

24. Официальный сайт ОИЯИ [Электронный ресурс] URL: https://jinr.ru/ (дата обращения 04.04.2022)

25. Официальный сайт проекта NICA [Электронный ресурс] URL: https://nica.jinr.ru (дата обращения 04.04.2022)

26. Ссылка на репозиторий проекта, https://github.com/Tims24/luminophore_detection/