**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

Описание предполагаемого способа решения

**по научно-исследовательской работе**

Тема: Реализация и исследование алгоритма генерации траекторий в динамической среде на основе OCTNet

|  |  |  |
| --- | --- | --- |
| Студент гр. 5303 |  | Губа Д.А. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2020

**Содержание**

[Постановка задачи 3](#_Toc58890290)

[Архитектура программной реализации 3](#_Toc58890291)

[Модуль предобработки 3](#_Toc58890292)

[Модуль обучения модели 3](#_Toc58890293)

[Модуль генератора траекторий 4](#_Toc58890294)

[UI модуль для отображения результатов 5](#_Toc58890295)

[Используемые технологии 5](#_Toc58890296)

[Данные для обучения 5](#_Toc58890297)

[Интерфейс для демонстрации работы 5](#_Toc58890298)

# Постановка задачи

В рамках научно-исследовательской работы необходимо реализовать алгоритм OCTNet и продемонстрировать правильность его работы. Алгоритм будет обучен на наборе карт с известными расстановками допустимых маршрутов, после чего будет протестирован на данных без разметки. После проверки корректности алгоритма, будет выполнено сравнение с конкурентами.

# Архитектура программной реализации

Архитектура программной реализации состоит из 4 модулей:

* Модуль предобработки данных
* Модуль обучения модели
* Модуль генератора траекторий
* UI модуль для отображения результатов

## Модуль предобработки

Программа будет загружать из сети dataset с набором данных для обучения и валидации модели. Данные в данном наборе не удобны для работы, поэтому необходимо провести конвертацию и валидацию данных. Поскольку набор данных представляет из себя файл с данными размером в несколько гигабайт, необходимо реализовать обработчик, который будет поддерживать кеширование и запоминать место остановки в случае ошибки. Предполагаемая модель парсера - потоковый парсер с указателем на последний считанный элемент набора данных.

## Модуль обучения модели

Архитектура модуля обучения изображена на рис. 1:

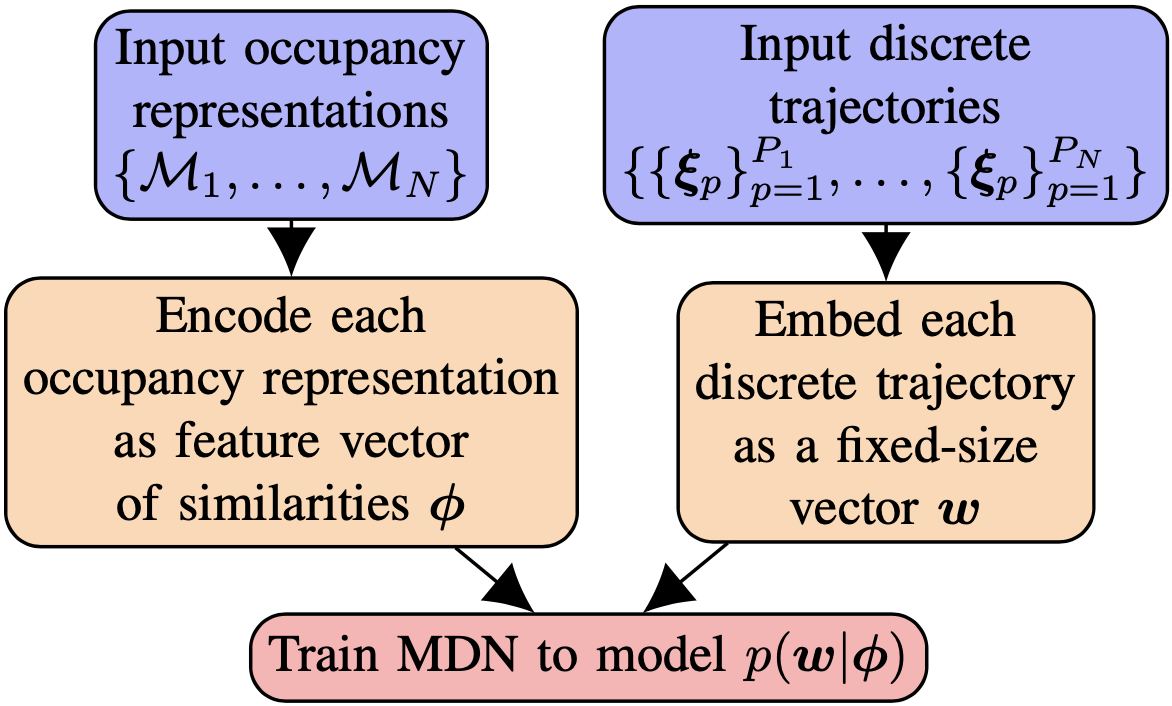


Рисунок 1 - архитектура модуля обучения модели

На вход модулю подается карта расстановки преград и коридоров, а также допустимые траектории. Модуль переводит данные в вектора признаков, которые подаются на вход алгоритму MDN для обучения модели.

## Модуль генератора траекторий

Архитектура модуля генератора траекторий изображена на рис. 2:

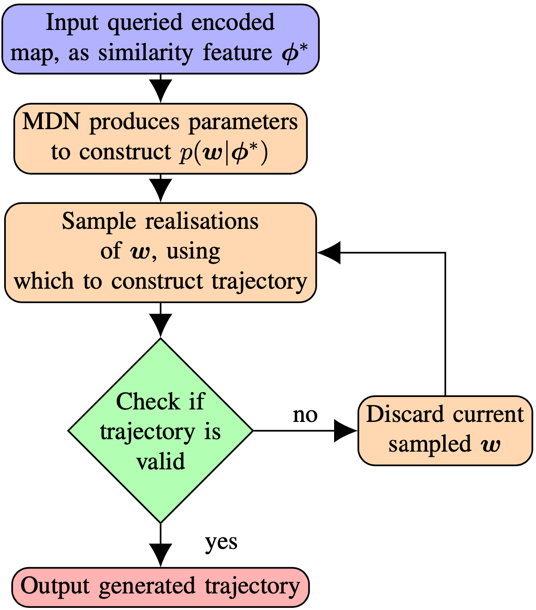


Рисунок 2 - архитектура модуля генератора траекторий

На вход подается вектор признаков, описывающий расстановку, по которому обученная модель умеет выдавать предполагаемую траекторию. Траектория может быть не корректной, поэтому модуль предполагает валидацию траектории с возможностью повторного перерассчета.

## UI модуль для отображения результатов

Данный модуль будет реализован при помощи библиотеки PyGTK. Модуль будет отвечать за отображение данных в репрезентативном для наблюдателя виде.

# Используемые технологии

Для разработки программного решения будут использованы:

* Язык программирования python3
* Фреймворк для машинного обучения TensorFlow
* REST API для получения данных
* PyGTK для отображения результатов

# Данные для обучения

Для обучения модели используется набор данных - occ-traj. Данный набор состоит из нескольких тысяч карт расстановок с размеченными маршрутами. Часть набора для обучения содержит расстановки для траекторий, а часть для тестирования расстановок для траекторий не содержит. Карта в наборе закодирована через нули и единицы, где 0 - зона заблокирована, 1 - зона свободна.

# Интерфейс для демонстрации работы

Интерфейс для демонстрации изображен на рис. 3:

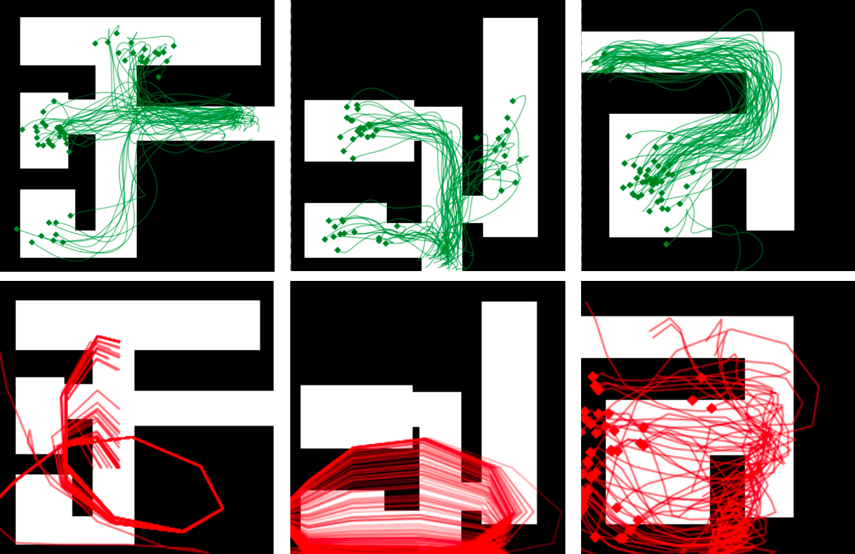


Рисунок 3 - интерфейс для демонстрации

Интерфейс будет отображать карту расстановки и построенные алгоритмом траектории. Корректные маршруты будут отмечены зеленым, а не корректные будут отмечены красным.