|  |  |
| --- | --- |
| **СОГЛАСОВАНО**  Доцент кафедры  ИАНИ ННГУ, к.ф.-м.н.  \_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Яшунин  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г. | **УТВЕРЖДАЮ**  Профессор кафедры  ИАНИ ННГУ, д.т.н.  \_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Старостин  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г. |

**Пояснительная записка**

**«Разработка модели нейронной сети для определения присутствия дронов на звуковой дорожке»**

**НИР «Drone sound detection»**

**(Шифр ПО «AFR»)**

Ответственный исполнитель

\_\_\_\_\_\_\_\_\_\_\_\_ Хромов Н.А.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

**Н. Новгород 2019**

Содержание

Введение3

Наименование программы3

Краткая характеристика области применения3

Основания для разработки3

Назначение разработки3

Назначение ПО «CNN Drone sound detection»3

Результат решения3

Требования к программному продукту3

Требования к функциональным характеристикам3

Требования к предварительным работам3

Требования к разрабатываемому ПО4

Общее описание алгоритма4

Назначение алгоритма4

Используемые библиотеки и модули4

Входные данные4

Описание алгоритма обучения рабочей версии нейронной сети 5

1. **Введение  
   1.1 Наименование программы:**Наименование ПО «CNN Drone sound detection»

**1.2 Краткая характеристика области применения**Объектом автоматизации является защита частной или государственной территории от несанкционированного вторжения. ПО «Drone sound detection» должно решать задачу по обнаружению дронов на основании звукового сигнала с определением их местоположения.

1. **Основания для разработки**Основание для выполнения ОКР: заказ конкретного потребителя.

Заказчик: Липкин С.

Исполнитель - Федеральное государственное автономное образовательное учреждение высшего образования «Нижегородский государственный университет им. Н. И. Лобачевского» (ННГУ)

Начало разработки - с момента заключения договора.

Окончание разработки - 30.06.2020

1. **Назначение разработки.**

**3.1 Назначение ПО «CNN Drone sound detection»**Модель нейронной сети предназначена для определения с присутствия звуков дронов на звуковой дорожке с определенной вероятностью.

**3.2 Результат решения**Результатом решения является обученная модель нейронной сети.

1. **Требования к программному продукту   
   4.1 Требование к функциональным характеристикам**Модель нейронной сети должна обеспечивать возможность выполнения перечисленных ниже функций:   
   1. Обнаружение присутствия дронов на звуковой дорожке.   
   2. Определение точности предоставленного ответа.

**4.2 Требования к предварительным работам**В рамках создания ПО «CNN Drone sound detection» должно быть выполнено следующее:

1. Формирование, поиск, модификация существующего датасета.

2. Ознакомление и выбор сущностей звукового сигнала, которые могут быть полезны для обучения сети.

**4.3 Требования к разрабатываемому ПО**В рамках создания ПО должно быть разработано следующее:

1. Архитектура нейронной сети CNN   
2. Обученная модель нейронной сети.

1. **Общее описание алгоритма**

**5.1 Назначение алгоритма**

Разработанный алгоритм служит для создания модели нейронной сети, способной дать ответ о наличии присутствия дрона на звуковой дорожке.

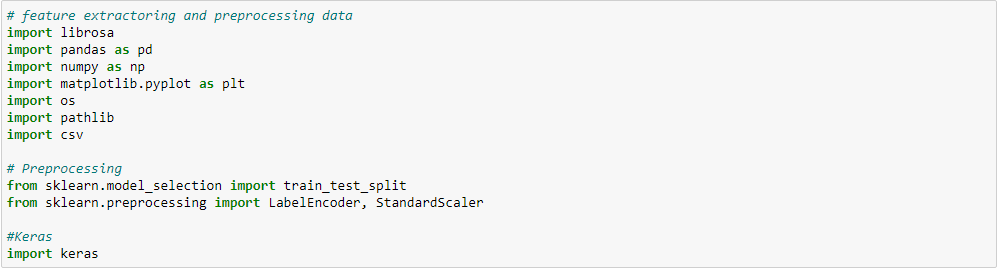
**5.1 Используемые библиотеки:**

* Librosa – библиотека для работы со звуком.
* Pandas – мощный пакет для анализа данных.
* NumPy – это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами.
* OS – модуль из стандартной библиотеки языка Python, испульзуется для работы с установленной ОС, а также файловой системой ПК.
* Matplotlib – Библиотека на языке программирования Python для визуализации данных двумерной (2D) графикой.
* Pathlib – Данный модуль упрощает работу с файлами и папками.
* CVS – Данный модуль даёт программисту возможность выполнять структурный анализ CVS файлов.
* Keras – открытая нейросетевая библиотека.

**5.2 Входные данные:**Для обучения и тестирования нейронной сети использовался датасет из открытого репозитория:   
<https://github.com/saraalemadi/DroneAudioDataset/tree/master/Binary_Drone_Audio>

1. **Описание алгоритма обучения рабочей версии нейронной сети**

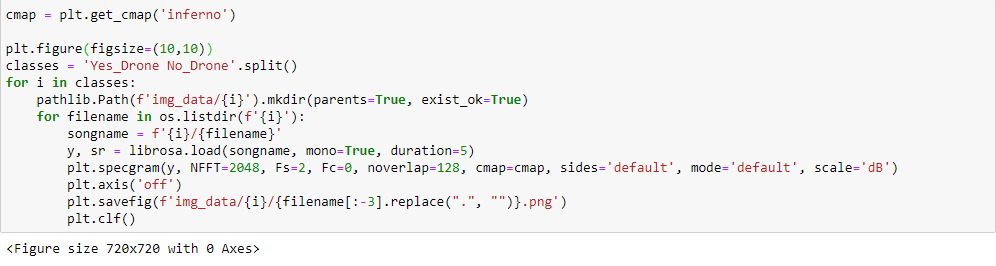
**Импорт библиотек:**



**Извлечение сущностей из каждой звуковой дорожки из датасета:**

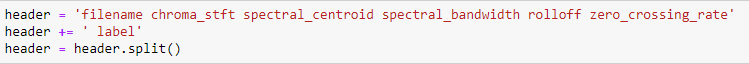
Датасет содержит 2 класса:   
1. Есть дрон (1332 звуковые дорожки)  
2. Нет дрона (1403 звуковые дорожки)

**Извлечение спектрограмм для каждой звуковой дорожки:**

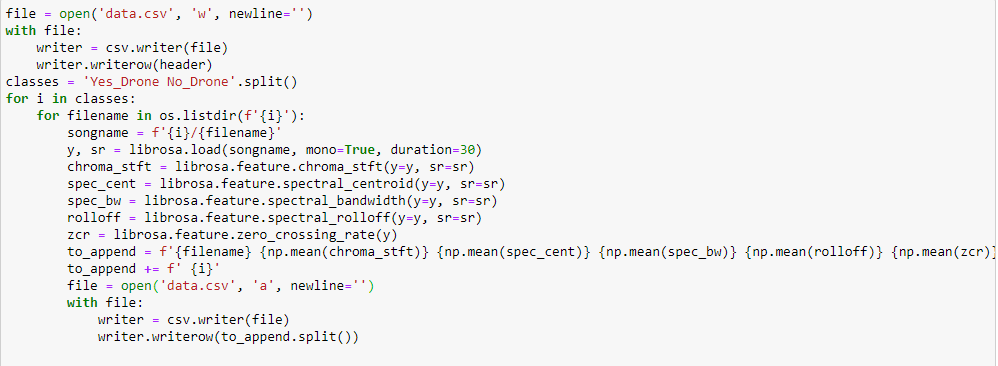


**Извлечение сущностей из спектрограмм**  
Мы будем извлекать:

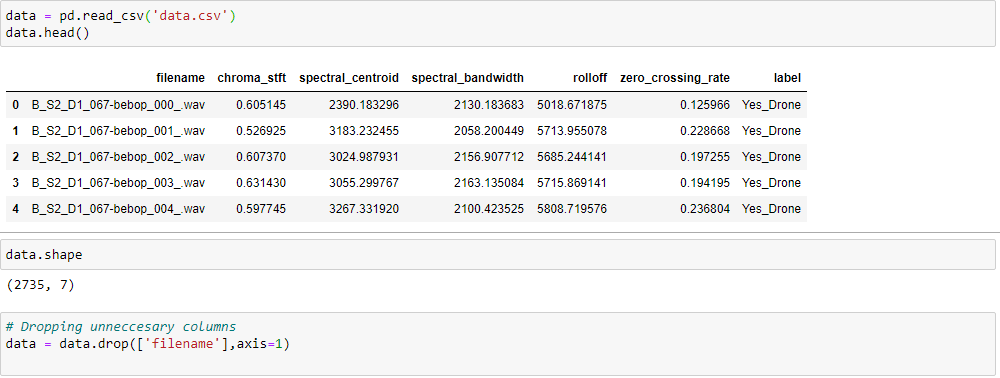
* Спектральный центроид
* Частота перехода через нуль
* Спектральная ширина
* Спектральный спад частоты
* Частота цветности



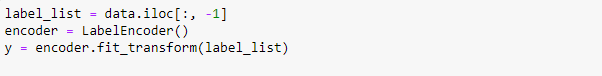
**Запись извлеченных данных в CVS файл:**



**Анализ данных с помощью Pandas:**



**Кодирование ярлыков**



**Предобработка (масштабирование) данных для обучения**

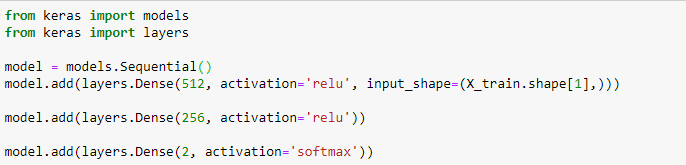


**Разделение данных на обучающий и тестовый набор**



**Построение модели нейронной сети**

**Настройка слоёв сети:**



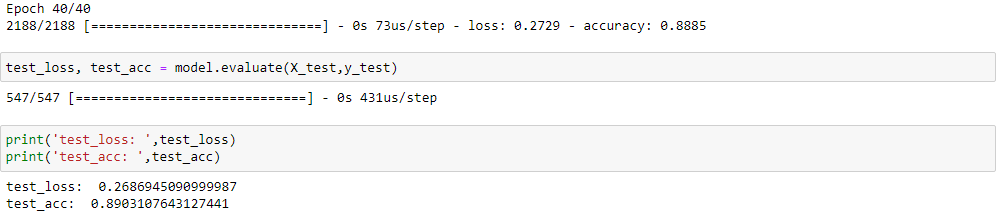
**Настройка конфигурации обучения:**

Прежде чем модель будет готова для обучения, нам нужно указать еще несколько параметров. Они добавляются на шаге compile модели:

* Функция потерь (Loss function) — измеряет точность модели во время обучения. Мы хотим минимизировать эту функцию чтоб "направить" модель в верном направлении.
* Оптимизатор (Optimizer) — показывает каким образом обновляется модель на основе входных данных и функции потерь.
* Метрики (Metrics) — используются для мониторинга тренировки и тестирования модели. Наш пример использует метрику accuracy равную доле правильно классифицированных аудиодорожек.



**Проверка модели на тестовых данных**



**Сохранение обученной модели и предобработчика данных**



В дальнейшем модель может быть использована для генерации предсказаний.

**Библиографический список**

[**https://www.tensorflow.org/tutorials/keras/classification?hl=ru**](https://www.tensorflow.org/tutorials/keras/classification?hl=ru)

[**https://gist.github.com/parulnith/7f8c174e6ac099e86f0495d3d9a4c01e#file-music\_genre\_classification-ipynb**](https://gist.github.com/parulnith/7f8c174e6ac099e86f0495d3d9a4c01e%23file-music_genre_classification-ipynb)