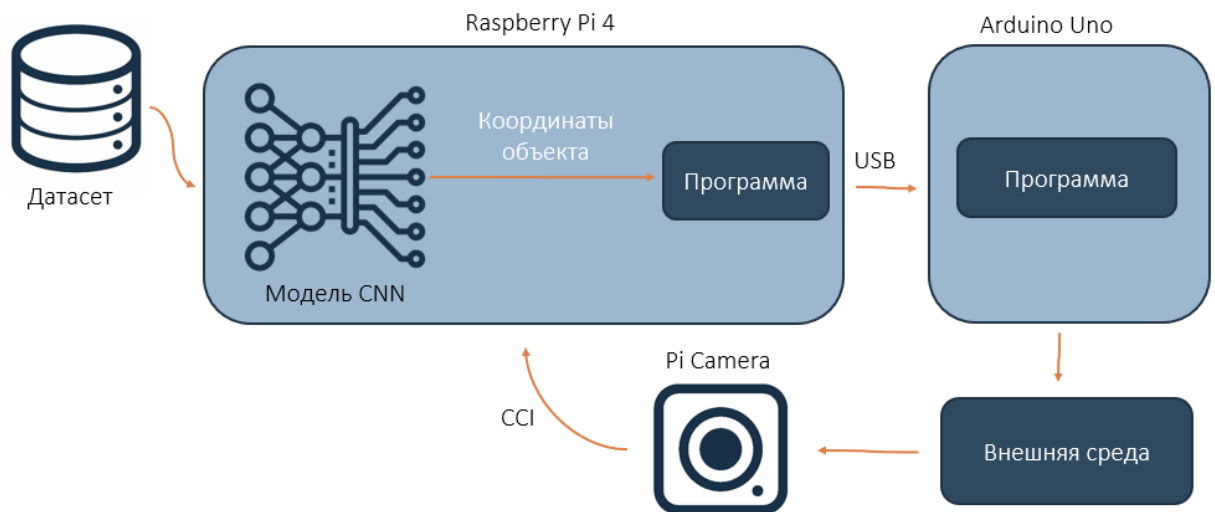


## Архитектура приложения



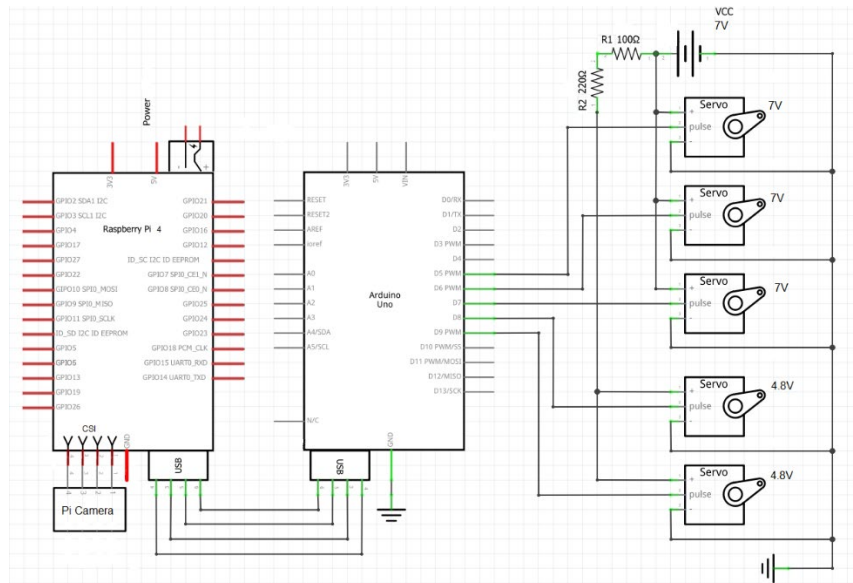
Система состоит из двух плат - Raspberry Pi 4 и Arduino Uno, соединенных через USB, камера подключена к Raspberry Pi 4 через интерфейс камеры.

После инициализации модели нейронной сети и камеры начинается первая итерация работы манипулятора: полученное фото внешней среды обрабатывается моделью CNN, которая определяет координаты распознанного объекта. Далее на основе данного массива точек происходит расчет углов поворотов, после чего сигнал поступает на Arduino, которая управляет пятью сервомоторами, манипулятор меняет свое положение в пространстве, цикл повторяется в разных плоскостях.

## Стек технологий



Для написания программы на Arduino Uno используется API Arduino, язык C++. Для программирования Raspberry Pi 4 использовалась Thonny Python IDE, Jupyter Notebook - для создания и тренировки нейронной сети. Разработка ведется на Raspbian - операционная система Raspberry Pi



Система состоит из двух плат - Raspberry Pi 4 и Arduino Uno, соединенных через USB, камера подключена к Raspberry Pi 4 через интерфейс камеры.

Питание Raspberry Pi 4 происходит через USB Type-C на 5 В, потребляемый ток до 3 А.

Питание Arduino Uno и пяти моторов происходит через дополнительный регулируемый источник питания PSN-305D, земля общая.

Т. к. сервомоторы имеют разные характеристики, было принято решение использовать шунтирующие резисторы, номиналы которых высчитываются исходя из формулы выходного напряжения, которое рассчитывается по формуле:

$$U_{\text{out}} = \frac{U_{\text{in}} * R_2}{R_1 + R_2}$$

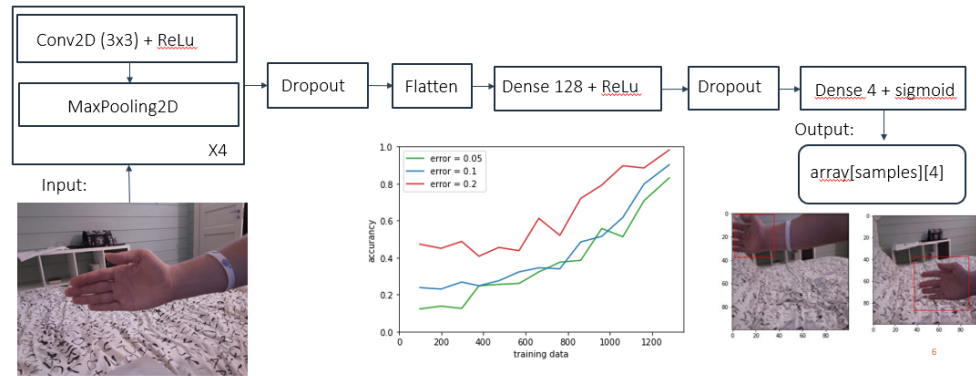
$U_{\text{out}}$  – выходное напряжение (= 4.8 В)

$U_{\text{in}}$  – входное напряжение (= 7.2 В)

$R_1, R_2$  – сопротивление резисторов

Таким образом,  $R_1 = 100 \text{ Ом}$ ,  $R_2 = 220 \text{ Ом}$

## Архитектура нейронной сети



Нейронная сеть состоит из четырех слоев свертки (Conv2D), осуществляющие операцию умножения матрицы изображения (28x28) на матрицу ядра свертки (3x3). Слои свертки чередуются с четырьмя слоями субдискретизации (MaxPooling2D), которые необходимы для сжатия размеров извлеченной карты признаков изображения. Слой прореживания (Dropout) используется для решения проблемы переобучения сети, далее преобразование в одномерный вектор (Flatten). потом идут полносвязный слой с линейной функцией активации, слой прореживания и последний полносвязный слой с сигмоидной функцией активации для классификации объектов

## Диаграмма классов

