Государственное бюджетное образовательное учреждение

“Школа 354 им. Дмитрия Михайловича Карбышева”

Российская Робототехническая Олимпиада

Творческий проект

Робот няня – “РобоКотик”

Выполнили:

Воля Максим

Девятов Леонид

Рынин Дмитрий

Руководитель:

Богачева Татьяна Петровна

Москва 2022г.

Содержание

1. Презентация команды

Здравствуйте, наша команда называется “RusTechno” и сейчас мы расскажем немного про неё. В нашей команде три человека: Воля Максим, Девятов Леонид и Рынин Дмитрий. Мы все учимся в одном классе и занимаемся робототехникой в школьном кружке. У нас есть распределение задач, между её членами, но по большей части мы принимаем все решения вместе. Например, то каким будет робот, его функционал и дизайн, мы решали все вместе. Но также каждый отвечает за часть. Например, Максим отвечал за программирование навигации робота и электроники. Лёня же отвечал за конструкцию, механику и тоже программирование электроники. А Дима отвечал за программную часть- развивающие игры, презентационные материалы. Но в то же время, каждый из нас любит узнавать что-то новое и поэтому, работаем мы вместе.

2. Идея проекта

В современном мире все больше и больше видна проблема свободного времени подрастающего поколения, дети предпочитают отдых в электронных гаджетах. Эксперты уверены, что большое количество экранного времени

взамен реальных предметов из мира вещей, может нанести серьезный вред психике ребенка. Всё дело в том, что родителям легче дать ребёнку планшет, вместо развивающей игры. Но это категорически неправильно, если игрушки заменяются гаджетами, снижается двигательная активность ребенка и не развиваются важнейшие навыки, что перерастает в неумение распоряжаться своим временем и планировать свой день.

Всё дело в том, что основа для развития психики — это мелкая и крупная моторика. Если ребенок проводит время с гаджетами, могут возникнуть оптико-пространственные нарушения. Малыш не научится правильно оценивать размер объектов, их расположение и удаленность друг от друга. В более старшем возрасте это отразится на учебе в школе, будут сложности со счетом, письмом, чтением и рисованием. Также у ребенка пострадает ощущение собственного «я».

Когда ребенок играет на планшете или смотрит мультики, происходит стимуляция мозговой активности. Ребенок оказывается в мире фантазий, понимает, что в реальной жизни не все так здорово и красочно, как в виртуальной. Получая удовольствие от сложных зрительных образов, мозг ребенка будет требовать еще и еще. Это может создать неравномерную нагрузку на психику и даже довести малыша до невроза.

Во избежание возникновения трудности в освоении бытовых навыков, общении со сверстниками и учебе в школе, ребёнку необходимо проводить время за игрой в игрушки.

Рис. 1 - Роль развивающих игр

3. Роль игры в кубики

На самом деле игра в кубики — это очень полезное и увлекательное занятие, которое развивает абстрактное мышление и наглядно показывает взаимодействие геометрических фигур. Все потому, что геометрические фигуры в таких наборах не ограничиваются только кубами, их может быть безграничное количество. Но зачастую дети не хотят сами играть в кубики, они любят повторять за взрослыми и в этом плане им интереснее играть с кастрюлями и другими предметами, используемыми родителями. Чтобы ребенок поиграл в кубики он должен видеть, что кто-то этим занимается и тогда ему станет интересно и он захочет повторить.

4. Цель проекта

Цель нашего проекта - разработать робота няню, способного играть с ребёнком в развивающие игры и помочь ему соблюдать распорядок дня, что невероятно важно в современном мире.

Данный робот отчасти позиционируется как замена гаджетам, то есть он будет проводить время с ребёнком, когда это не могут делать родители. В его обязанности входят такие пункты, как игры с ребёнком в развивающие игры, перечень которых выбирают родители, что позволяет выбрать игры именно под возраст ребёнка. А также робот может делать физические упражнения и попросить ребёнка повторить их. Перечень игр и расписание занятий на день, родители могут настроить на сайте, а также наблюдать, соблюдает ли ребенок его.

5. Этапы разработки проекта

Работа над проектом началась с проработки идеи, потому что идей было много, но в итоге мы все склонились к этой. После принятия идеи нужно было хорошо разобраться в проблематике, чтобы понимать, каким функционалом должен обладать робот и как он должен выглядеть. Как раз по поводу вида, мы долго думали над дизайном робота, как сделать его таким, чтобы ребёнок не испугался, а хотел поиграть с ним. В итоге мы пришли к виду котика, который бы мог напомнить ребёнку доброго персонажа из мультиков и просто внушал доверие. Проработав дизайн, нужно было решить с управляющими модулями и электронной начинкой. А также придумать как разместить всё внутри робота, в этом нам помогло 3D моделирование, с помощью него были сделаны модели корпуса робота, а на нём уже продуманно, куда, что поставить. После проработки всех частей робота мы приступили к его созданию. Всё началось с постройки каркаса робота, на который в последствии была надета мягкая оболочка. Была разведена и получена плата для движения робота и управления всеми сенсорами. Параллельно с этим велась работа над сайтом и программой обучения для ребенка.

Работа по месяцам

Таблица 1

Месяц	Проделанная работа
Январь	Появление идеи, изучение темы развивающих игр и продумывание функционала робота.
Февраль	Итоги по функционалу, продумывание дизайна робота и начало программирования обучающих программ.
Март	Создание каркаса, написание программы для игры в кубики, разработка платы для управления роботом.

Апрель	Пайка платы, программирование движения, начало работы над программой, обучающей грамоте.
Май	Создание 3D модели головы робота, печать деталей для мягкой оболочки,
Июнь	Шитьё оболочки робота, программирование электроники робота, работа над пояснительными материалами.

6. Презентация роботизированного решения

6.1 Выбор идеи

Разумеется, у нас не возникла изначально именно это идея, были ещё идеи, которые в итоге мы не приняли. Мы так же рассматривали вариант робота, для помощи в больницах людям с болезнью Альцгеймера, но разобравшись в этой теме, поняли, что таким людям нужно именно внимание, человека, а не робота. Так же была идея сделать небольшую плюшевую игрушку, для помощи детям с домашними заданиями, именно тогда зародилась идея работать именно для детей. У многих сейчас есть няни, почему бы ей не быть роботом? Поскольку тема работы с детьми требует много знаний в этой области, а у нас их нет, мы пошли в детский сад нашего школьного комплекса, чтобы попросить советы и пожелания, от людей, которые постоянно работают с детьми именно нашей возрастной категории. А также, чтобы узнать о роли развивающих игр из первых уст и сформировать образ робота, который подошёл бы для детей.

6.2 Аналоги

Так же стоит сказать про аналоги, которые, конечно, есть. Но зачастую это просто игрушки, с которыми ребёнок может играть. Или полноценные роботы няни, мы рассмотрели каждую категорию.

Рис. 2 – Робот Anki Vector

На этом рисунке, представлен робот Vector, это робот, размеры которого настолько малы, что он умещается на ладони. Он может помочь с поиском информации в интернете, заказе товаров, а также управлении домашней техникой. Но по большей части, главная его функция — это общение и игры с владельцем. Ещё у робота имеется HD камера, с помощью которой, он может перемещаться, избегая препятствий, делать фото и распознавать людей. Но в отличии от нашего робота, он предназначен больше для роли умного помощника и развлечения.

Рис. 3 – Робот Jibo

На этом рисунке, изображён робот Jibo, эта "говорящая голова" способна не только поддерживать разговор, но и автоматически фотографировать членов семьи или записывать их на видео, работать персональным секретарем с голосовым управлением, контролирующим ваши сообщения и напоминания, а также рассказывать детям истории.

Помимо таких решений, есть ещё одно – это AvatarMind iPal, он предлагает облегчить труд родителя и взять вопросы воспитания на себя. Робот рассчитан на детей в возрасте от трех до восьми лет. Одной зарядки робота хватит на присмотр за детьми в течение пары часов. Создатели робота предполагают, что воспитание прячется где-то между играми и уроками, потому робот запрограммирован на прохождение элементарной образовательной программы и на игры. Кроме игр робот контролирует детей, предупреждает их об опасности, измеряет, а также фотографирует и снимки скидывает родителям.

Рис. 4 – Робот AvatarMind iPal

6.3 Механика робота

Основной частью робота является механика, мы разделили ее на несколько частей. Первая - ходовая, она включает в себя 4 мотора постоянного тока 12 вольт от конструктора TETRIX, это сделано, потому что мы используем всенаправленные omni колёса, которые позволяют нам ездить во всех направлениях и тем самым робот не будет мешать в квартире и станет более многофункциональным с точки зрения движения. Второй частью является голова, эта часть является самой важной, с точки зрения коммуникации, потому что именно с головой ребёнок будет постоянно общаться. Голова напечатана на 3D принтере, а мордочка сделана с помощью адресной светодиодной матрицы, чтобы мы могли сделать мимику глаз и рта,

что будет хорошо влиять на игру с роботом. Третья часть же — это манипуляторы. Они тоже крайне важны для игры с роботом, но особенно в физических упражнениях, без них это было бы невозможно. Манипуляторы имеют несколько степеней свободы которые достигнуты путём соединения сервоприводов на нескольких осях координат, они могут сгибаться в локтях, что ещё добавляет возможностей в упражнениях. (См. Приложение А)

6.4 Электроника проекта

Помимо механики, важную часть в нашем проекте играет электроника, она управляет всем роботом. Специально для этого робота, мы разработали плату управления, она находится в туловище робота и охлаждается вентиляторами, которые стоят в его нижней части. Плата была сделана в программе KiCad и заказана в компании Резонит. Плата содержит: Arduino Mega Pro mini, она управляет всей логикой движений робота и общается с raspberry, которая в свою очередь работает с машинным зрением и нейронной сетью, 4 драйвера для управления моторами, стабилизатор напряжения, потому что моторы потребляют 12 вольт постоянного тока, а Arduino требуется только 5, так же выведены отдельное контакты для управления сервоприводами и подсветкой на лице робота. А также ещё ряд компонентов, которые нужны для работы платы, например резисторы и конденсаторы разных номиналов, продумана система охлаждения. (См. Приложение Б)

6.5 Программная часть

Последняя часть проекта – это программная часть, в неё входит, как и программа для движения робота, так и нейронные сети для работы с машинным зрением и обработки игр, физических упражнений и других активностей ребёнка. Поскольку, в роли главного микроконтроллера мы используем Arduino, программирование движения происходило на языке C++ я среде программирования Arduino IDE. Код для работы нейронной сети и машинного зрения, мы писали на языке Python, далее будет подробнее рассказано про работу нашей нейронной сети. (См. Приложение Г)

6.6 Нейронные сети

Нейронные сети представляют собой самообучающиеся модели, имитирующие деятельность человеческого мозга. Они способны не только выполнять однажды запрограммированную последовательность действий над заранее определенными данными, но и сами анализировать вновь поступающую информацию.

Основным достоинством нейронных сетей является возможность эффективно строить нелинейные зависимости, более точно описывающие наборы данных по сравнению с линейными методами статистики.

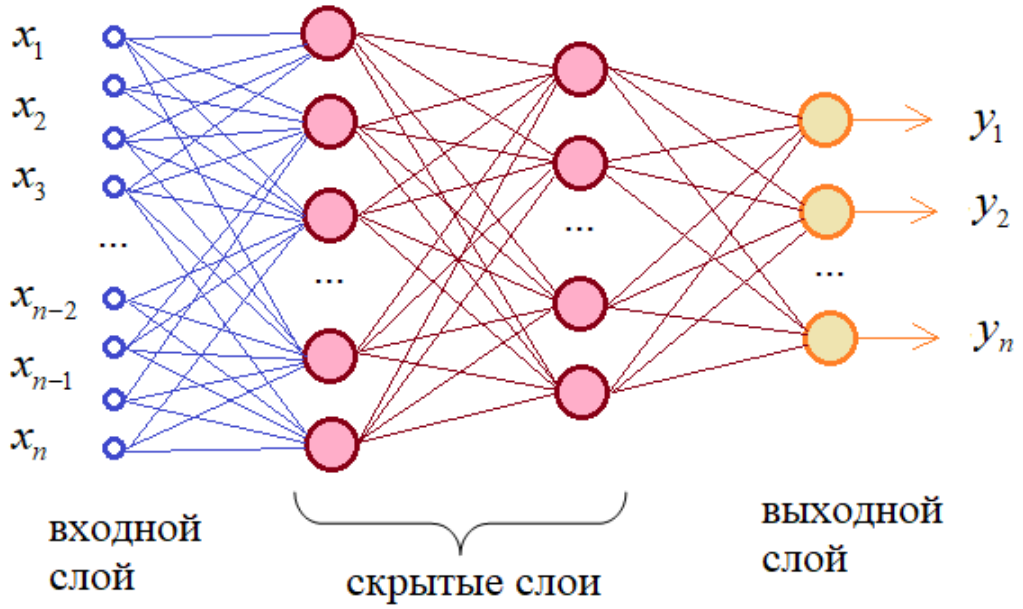

Рис. 5 – Схематичное изображение стандартной нейронной сети

В качестве образов могут выступать различные по своей природе объекты: символы текста, изображения, образцы звуков и т. д. При обучении сети предлагаются различные образцы образов с указанием того, к какому классу они относятся. Образец, как правило, представляется как вектор значений признаков. При этом совокупность всех признаков должна однозначно определять класс, к которому относится образец. В случае, если признаков недостаточно, сеть может соотнести один и тот же образец с несколькими классами, что неверно. По окончании обучения сети ей можно

предъявлять неизвестные ранее образы и получать ответ о принадлежности к определённому классу.

Метод обратного распространения ошибки - метод вычисления градиента, который используется при обновлении весов. Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении обратном прямому распространению сигналов в обычном режиме работы

Градиентный спуск: $W^{t+1} = W^t - a * dE/dW(W^t)$

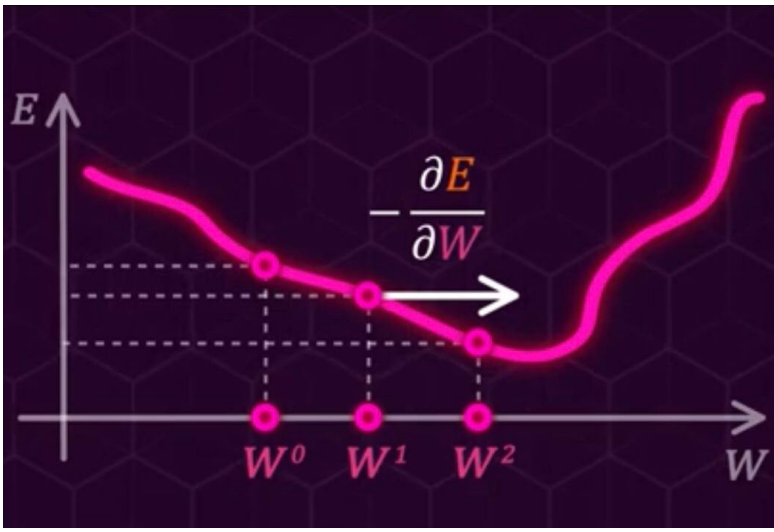

Рис. 6 – Градиентный спуск

Рис. 7 – Обратное распростанение ошибки

Градиент: $dE/DW = \{dE/dW^1, dE/dW^2, dE/dW^3 \ldots\}$

```
py -3.7  ney_buckv.py

Found 8147 images belonging to 4 classes.
Found 0 images belonging to 0 classes.
Found 24 images belonging to 4 classes.
  ...
    to
  ['...']
Train for 8147 steps
Epoch 1/6
 515/8147 [>.............................] - ETA: 1:11:54 - loss: 1.4512 - accuracy: 0.5049
```

Рис. 8 – Пример отладочной информации в момент обучения

Произошла одна эпоха (epoch) — весь “датасет” прошел через нейронную сеть в прямом и обратном направлении только один раз.

Почему используется более одной эпохи?

Вначале не понятно, почему недостаточно одного полного прохода датасета через нейронную сеть, и почему необходимо пускать полный датасет по сети несколько раз.

Нужно помнить, что мы используем ограниченный датасет, чтобы оптимизировать обучение и подстроить кривую под данные. Делается это с помощью градиентного спуска — итеративного процесса. Поэтому обновления весов после одного прохождения недостаточно.

Одна эпоха приводит к недообучению, а избыток эпох — к переобучению:

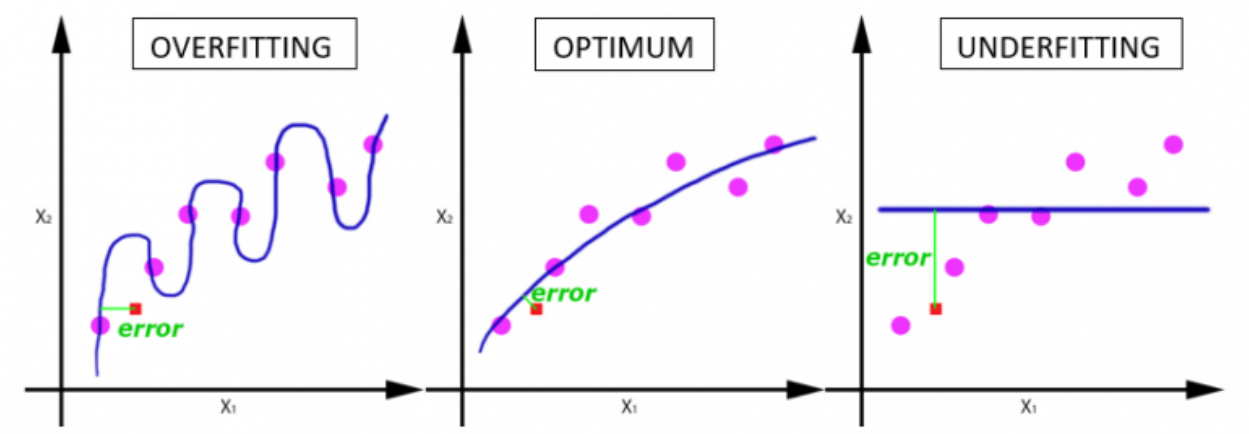

Рис. 9 – Обучение нейронной сети и эпохи

С увеличением числа эпох, веса нейронной сети изменяются все большее количество раз. Кривая с каждый разом лучше подстраивается под данные, переходя последовательно из плохо обученного состояния (последний график) в оптимальное (центральный график). Если вовремя не остановиться, то может произойти переобучение (первый график) — когда кривая очень точно подстроилась под точки, а обобщающая способность исчезла.

В нашем проекте активно используются нейронные сети. Рассмотрим один из примеров их использования. Нейронная сеть помогает «робоКотику» играть с ребёнком. Робот показывает сооружение из кубиков, которое малыш

должен попытаться повторить, после чего он получит ответ: верно собрана конструкция или нет. Для этого используется **свёрточная нейронная сеть**

Свёрточная нейронная сеть — специальная архитектура искусственных нейронных сетей, нацеленная на эффективное распознавание образов, входящая в состав технологий глубокого обучения. Использует некоторые особенности зрительной коры, в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток. Таким образом, идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв и субдискретизирующих слоёв. Структура сети — принципиально многослойная, без обратных связей. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки.

Название архитектура сети получила из-за наличия операции свёртки, суть которой в том, что каждый фрагмент изображения умножается на матрицу свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

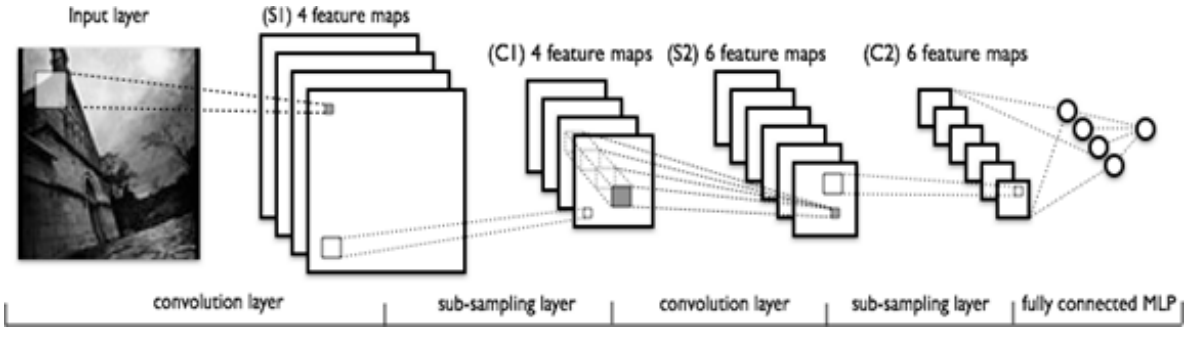

Рис. 10 – Свёрточные нейронные сети

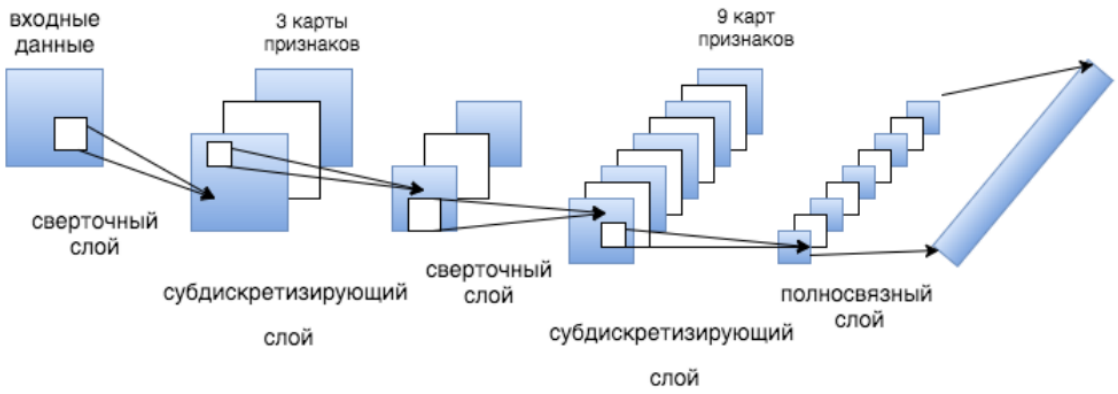

Рис. 11 – Свёрточные нейронные сети

6.7 Корпус робота

После общения с воспитателями в детском саду, мы пришли к выводу, что корпус робота должен быть достаточно мягким, чтобы ребёнок мог обнять робота или просто контактировать с ним было приятно. Некоторые части, которые имеют внутри электронику, должны быть защищен, поэтому находится в корпусе, который мы напечатали на 3D принтере из пластика PLA. Это экологичный, биоразлагаемый пластик, получаемый из натурального сырья. Чтобы котик был мягким, мы решили сделать упругий каркас вокруг алюминиевого. Это мы смогли сделать с помощью сантехнических труб, на которые в последствии был натянут чехол, сшитый нами. (См. Приложение В)

6.8 Социальное взаимодействие и инновации

При сравнении нашего робота с аналогами мы выяснили, что роботов, выполняющих те же функции и сочетающих в себе привлекательный внешний вид, просто нет. При рассмотрении внедрения нашего робота в производство и продажу, мы можем предоставить покупателем выбор. Поскольку мягкая оболочка одевается на робота, вы сможете купить робота в разных обличиях, например зайца или собаки, помимо стандартного кота или докупить потом отдельно чехол. Так же пользователи смогут докупать развивающие игры на сайте, дети очень быстро растут и стандартного пакета, рассчитанного на вест курс, может быть недостаточно или родители захотят купить дополнительные игры. (См. Приложение Г)

Приложения

Приложение А

Механика робота

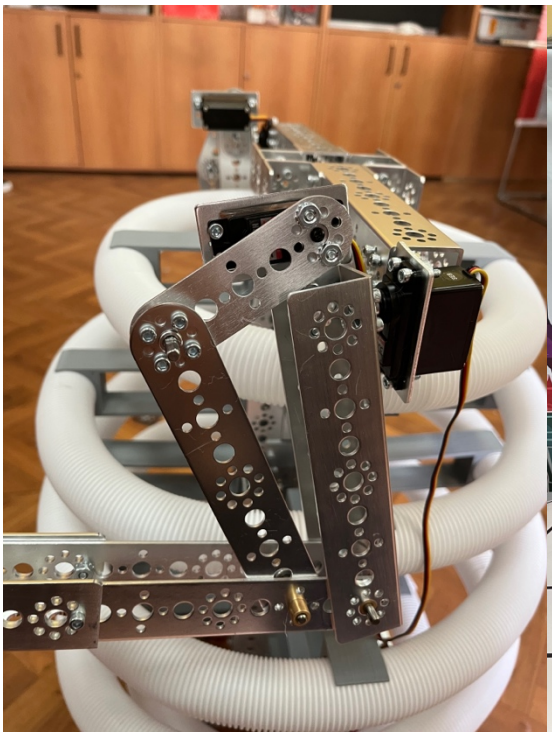

Рис. 12 – Кинематика манипулятора Рис. 13 - Строение робота

Приложение Б

Электроника проекта

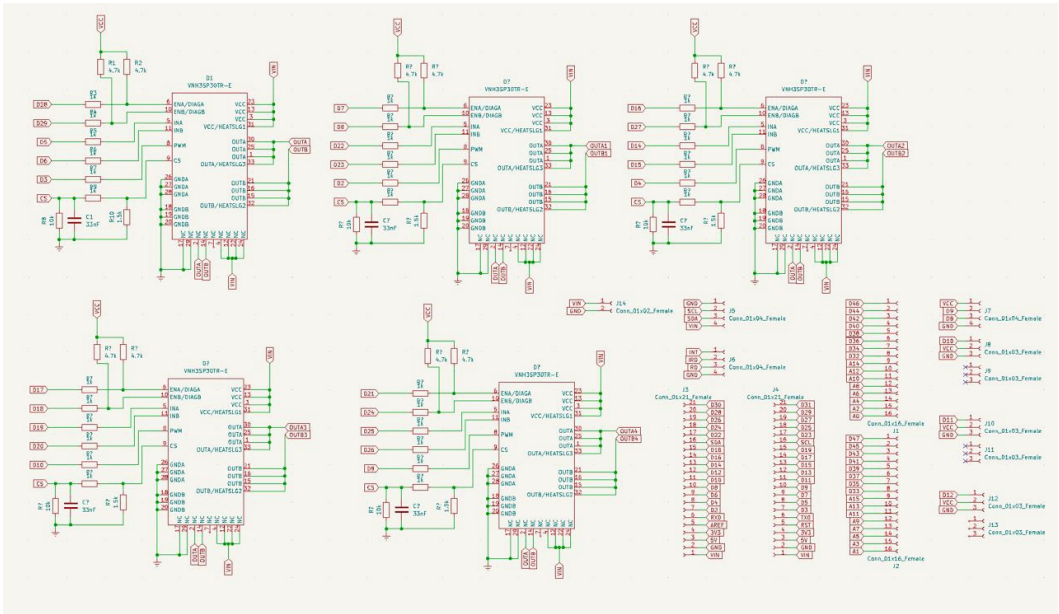

Рис. 14 – Принципиальная схема платы

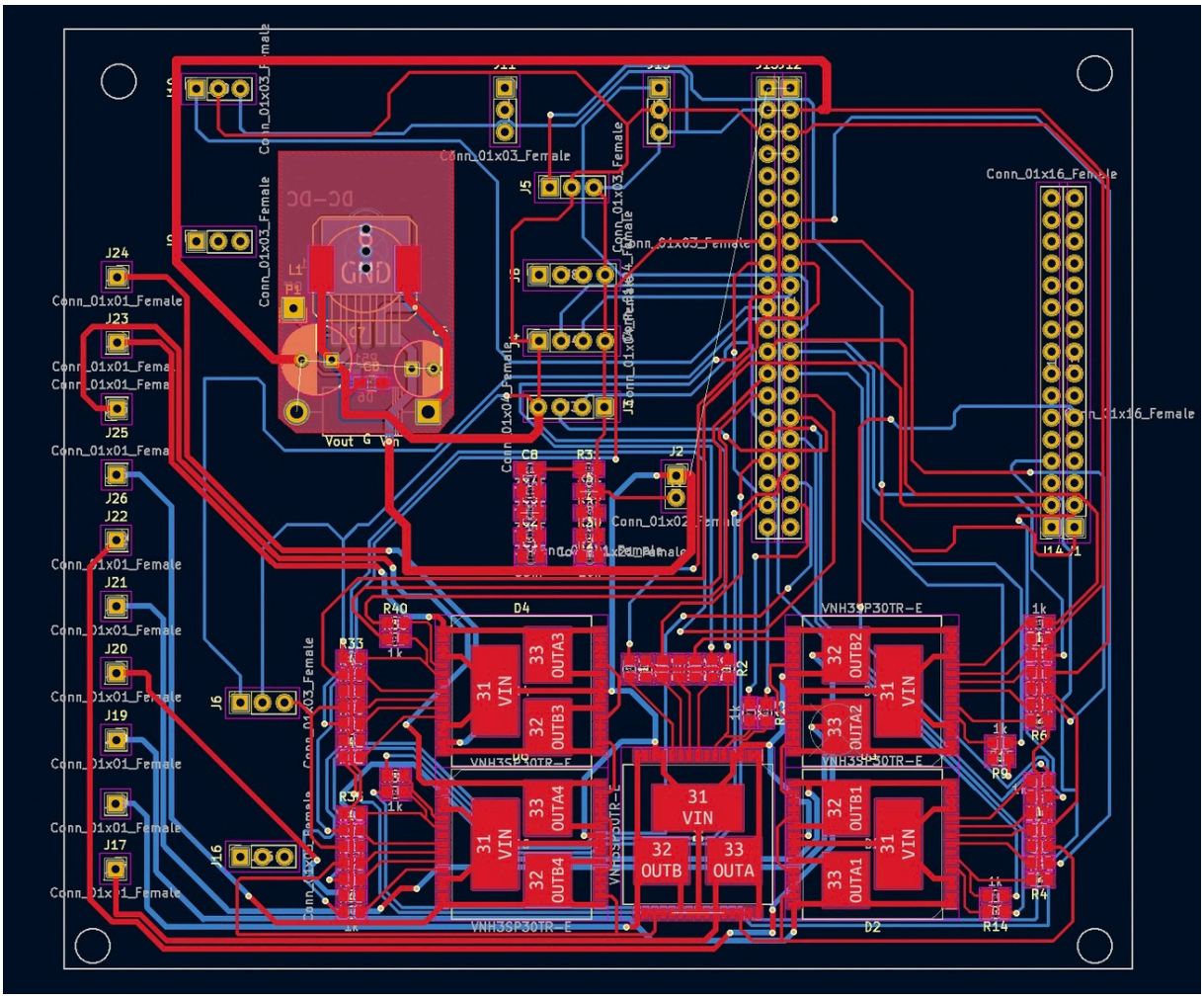

Рис. 15 – Трассировка печатной платы в программе KiCad

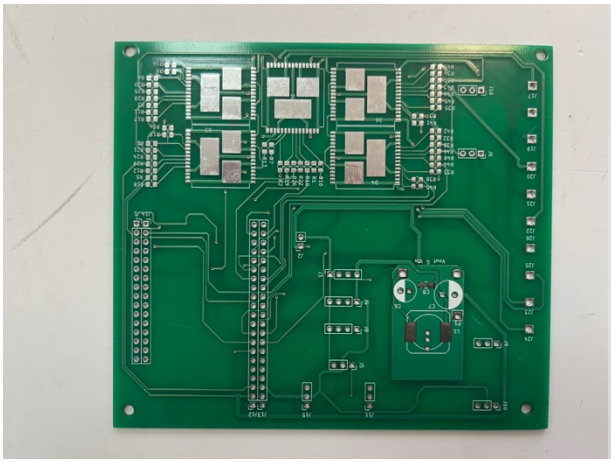

Рис 16. – Плата без компонентов

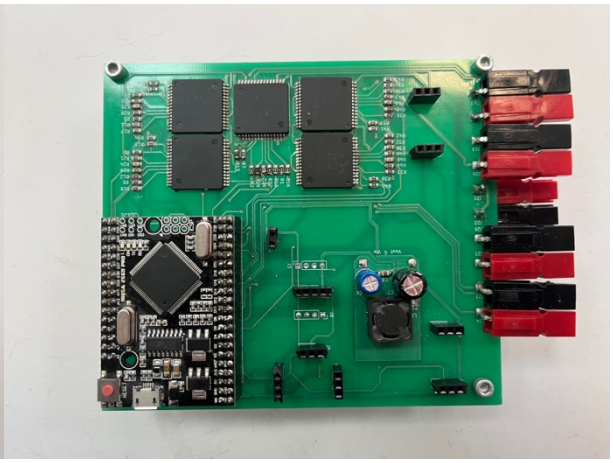

Рис. 17 – Плата с компонентами

Приложение В

Корпус робота

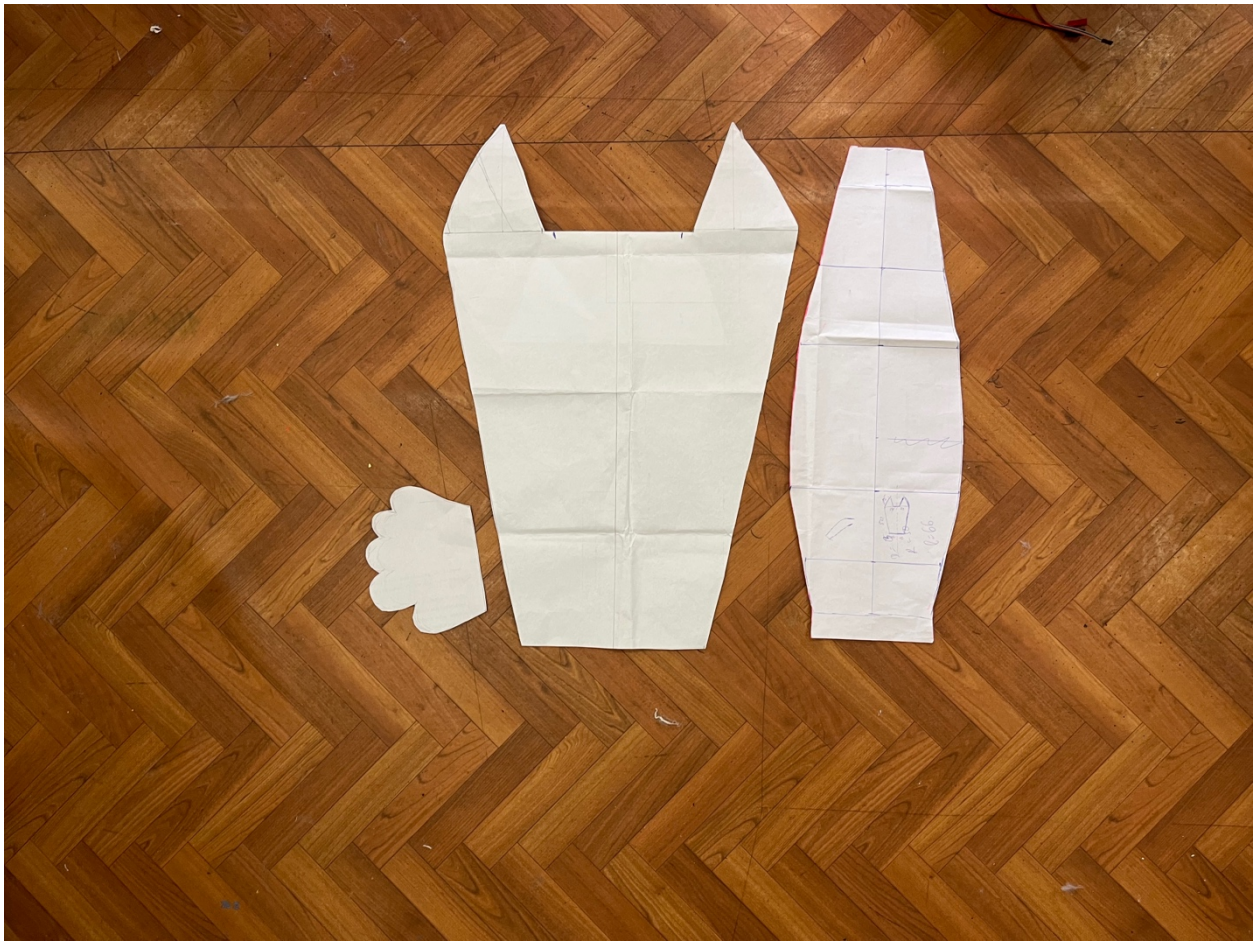

Рис. 18 – Выкройка для оболочки

Приложение Г.

Программа игры в кубики.

```
import string,cgi,time
from os import curdir, sep
from http.server import BaseHTTPRequestHandler, HTTPServer from urllib.parse import
unquote

import pygame
import random
from tensorflow.keras.models import*
```

```
from tensorflow.keras.preprocessing import image import tensorflow as tf
import numpy as np
import PIL
from PIL import Image
import cv2
import time

import requests
from threading import Thread
import urllib.parse
from urllib.parse import urlparse
import json
info = 0
class MyHandler(BaseHTTPRequestHandler):

def do_GET(self): global info global j_vern global j_nevern try:

url = self.path
self.send_response(200)
self.send_header('Content-type', 'text/html') self.send_header('Access-Control-Allow-Origin',
'*') self.send_header("Access-Control-Allow-Methods" , "GET,POST,PUT,DELETE,OPTIONS")
self.send_header("Access-Control-Allow-Headers", "Content-Type, Access-Control-Allow-

Headers, Authorization, X-Requested-With") self.end_headers()

if (url.find("timetable") > -1): info = 0

s = unquote(url)
print(s)
tb7 = s[s.find("tb7")+4:s.find("tb8")-1] tb8 = s[s.find("tb8")+4:s.find("tb9")-1] tb9 =
s[s.find("tb9")+4:s.find("tb10")-1] tb10 = s[s.find("tb10")+5:s.find("tb11")-1] tb11 =
s[s.find("tb11")+5:s.find("tb12")-1] tb12 = s[s.find("tb12")+5:s.find("tb13")-1] tb13 =
s[s.find("tb13")+5:s.find("tb14")-1] tb14 = s[s.find("tb14")+5:s.find("tb15")-1]

tb15 = s[s.find("tb15")+5:s.find("tb16")-1] tb16 = s[s.find("tb16")+5:]
print('7:00 -', tb7)
print('8:00 -', tb8)

print('9:00 -', tb9) print('10:00 -', tb10) print('11:00 -', tb11) print('12:00 -', tb12) print('13:00 -',
tb13) print('14:00 -', tb14) print('15:00 -', tb15) print('16:00 -', tb16)

elif (url.find("getinfo") > -1): if info == 1:

self.wfile.write(bytes("1", "utf-8" )) else:

self.wfile.write(bytes("0", "utf-8" )) print(info)
print('otvet otpravlen')

elif (url.find("newinfo") > -1): info = 1
```

```
return
except IOError:

self.send_error(404,'File Not Found: %s' % self.path)

def server(): try:

server = HTTPServer(('', 8080), MyHandler) print ('started httpserver...') server.serve_forever()

except KeyboardInterrupt:
print ('^C received, shutting down server') server.socket.close()

def kub():
with open('neyron3.7.json', 'r') as f:

model = model_from_json(f.read()) model.load_weights('neyron3.7.h5')# экспортируем
нейронную сеть classes = ['n','n','n','n','n','vern']
pygame.init()
W,H = 500,500
win = pygame.display.set_mode((W,H))
FPS = 90

clock = pygame.time.Clock() nevern = False

verno = False a = 50
b = 50

neverno_img = pygame.transform.scale(pygame.image.load('неверно.png').convert_alpha(),
(100, 100))

verno_img = pygame.transform.scale(pygame.image.load('верно.jpg').convert_alpha(), (100,
100))

while True:
for event in pygame.event.get():

if event.type == pygame.QUIT: exit()

press = pygame.mouse.get_pressed() pos = pygame.mouse.get_pos()
keys = pygame.key.get_pressed()
# ------------------------

if verno == True:
win.fill((0,0,0))
win.blit(verno_img, (200, 200))
win.blit(pygame.font.Font(None, 36).render('верно', 1, (0, 180, 0)), (215, 50)) p =
random.randint(0,3)

p1 = random.randint(0,2)
timer = 0
```

```
verno = 0
r = requests.get('http://127.0.0.1:8080/newinfo?')

# ------------------------- if nevern == True:

win.fill((0,0,0))
win.blit(neverno_img, (200, 200))
win.blit(pygame.font.Font(None, 36).render('неверно', 1, (180, 0, 0)), (200, 50)) timer = 0
nevern = 0

if keys[pygame.K_SPACE]: win.fill((0,0,0))
cap = cv2.VideoCapture(1) ret, img = cap.read() cv2.imwrite('img.jpg',img)

img_path = 'img.jpg'# экспортируем изображение

img = image.load_img(img_path, target_size =(300,300))# загружаем его, приводя к нужным размерам(кол-во пикселей, использ. при обчение сети)

img = np.asarray(img.convert('RGB'))# преобразуем загруженное изображение в 3х- мерный масив numpy

x = image.img_to_array(img) x /= 255
x = np.expand_dims(x, axis=0)

prediction = model.predict(x)# получаем информацию о процентаже вероятностей, что на фотографие тот или иной обьект(один из категорий)

x = prediction[0]# из 2х-мерного в 1-мерный print(x)

g = (classes[np.argmax(prediction)]) print(g)

if g == 'vern':
verno = 1
nevern = 0
r = requests.get('http://127.0.0.1:8080/newinfo?')

else:
verno = 0 nevern = 1

pygame.display.update() clock.tick(120)

f_server, f_kub = Thread(target = server), Thread(target = kub)# создание параллейных потоков

if __name__ == '__main__': f_server.start(), f_kub.start() f_server.join(), f_kub.join()
```

Приложение Д

Сайт для родителей