

Приложение  
к условиям проведения заключительного  
этапа Республиканского конкурса  
технического творчества учащейся  
молодежи «Инженеры будущего»

Конкурсное задание номинации  
«Искусственный интеллект»

Искусственный интеллект (ИИ) – это область компьютерных наук, изучающая вопросы, связанные с разработкой систем, имитирующих человеческие когнитивные способности, такие как обучение, рассуждение, решение проблем и принятие решений.

Машинное обучение – это область искусственного интеллекта, определяющая набор методов, техник и парадигм, которые позволяют системе обучаться на прецедентах и выявлять закономерности, делая прогнозы без явного программирования.

Глубокое обучение – это область машинного обучения, которая использует многослойные нейронные сети для анализа данных и принятия решений.

Участники привозят самостоятельно:

ноутбук с дискретной видеокартой, пригодной для работы с алгоритмами машинного обучения, в т.ч. для применения методов глубокого обучения, веб-камерой и микрофоном (допускаются встроенные), а также программным обеспечением, необходимым для выполнения конкурсного задания;

флеш-диск для копирования тестовых данных.

Финальное тестирование и запуск приложения выполняется на компьютере (ноутбуке) с дискретной видеокартой, веб-камерой и микрофоном, предоставляемые оргкомитетом и параметры которого сообщаются участникам в начале конкурса. Также оргкомитет информирует участников, какое разрешение видеопотока с камеры будет использоваться при тестировании, а также какие ограничения на ресурсы должны быть (используемый объем оперативной памяти, память GPU, объем жесткого диска).

Перед началом конкурса оргкомитет готовит тестовые данные:

папка “train”, в которой находятся изображения для обучения модели (с сохранением структуры папок), на которых участник разрабатывает и тестирует модель нейронной сети;

папка “test”, в которой находятся изображения из тестового набора и которые будут использоваться для “слепого” тестирования.

Участники копируют папку “train” на свой флеш-диск в начале конкурса.

На ноутбуке рекомендуется предварительно установить и настроить:  
интерпретатор языка программирования Python 3.x;

минимальный набор библиотек актуальных версий для работы с алгоритмами машинного обучения (PyTorch либо TensorFlow (на выбор), scikit-learn, ...);

библиотека NumPy для работы с матричными/векторными вычислениями;

библиотека Pandas для работы с данными;

библиотека Matplotlib для визуализации;

библиотека OpenCV с настроенным доступом из Python;

библиотека PyQt5 для создания интерфейса пользователя;

библиотека Whisper и/или VOSK для обработки голоса;

среда разработки (Microsoft Visual Studio Code, PyCharm,);

расширение к Microsoft Visual Studio Code для доступа к Google Colab для ускорения обучения и работы нейронных сетей (при необходимости).

В конкурсе принимают участие учащиеся в возрасте 14-18 лет (включительно).

Конкурс состоит из одного задания по разработке приложения с использованием алгоритмов машинного обучения, в т.ч. нейронных сетей. Вариант задания определяется случайным образом.

На выполнение конкурсного задания отводится:

8 (восемь) часов;

Исходные данные для вариантов заданий определяются экспертами по компетенции перед началом проведения соревнований заключительного этапа республиканского конкурса технического творчества учащейся молодежи «Инженеры будущего».

Команда за отведенное время должна предоставить готовое приложение, оцениваемое по следующим критериям:

#### 1. Дизайн-проект

Начисление баллов выполняется за корректную и качественную реализацию следующих пунктов:

общее описание работы приложения;

блок-схема алгоритма работы приложения;

схемы всех экранов приложения, в т.ч. переходы между ними;

#### 2. Прототип

Начисление баллов выполняется за корректную и качественную реализацию следующего функционала:

реализация основных функций задания;

- соответствие дизайн-проекту;
- запуск на целевом устройстве;
- обеспечение приемлемой скорости обработки видеопотока (либо скорость обработки видеопотока - не менее 1 кадра в секунду (1 FPS));
- оценка точности (accuracy) базовой модели нейронной сети на тестовых данных;
- построить ROC-кривую;
- поддержка управления приложением с помощью голоса (русский и/или английский язык). При наличии требования в конкурсном задании.

### 3. Оптимизация

Начисление баллов выполняется за корректную и качественную реализацию следующих пунктов:

- использование аугментации данных;
- дообучение (fine-tuning) готовой модели для решения конкурсного задания;
- оценка точности (accuracy) модели нейронной сети на тестовых данных;
- построить ROC-кривую;
- построение матрицы ошибок (confusion matrix);
- оценка метрики F1;

### 4. Сборка и запуск проекта

Начисление баллов выполняется за корректную и качественную реализацию следующих пунктов:

- корректный запуск собранного проекта на целевом устройстве (окно приложения с демонстрацией видеопотока с веб-камеры);
- отсутствие «падений» во время работы приложения;
- наложение результата работы алгоритма машинного обучения поверх видеопотока;
- оценка быстродействия распознавания команд с помощью голоса. При наличии требования в конкурсном задании.;
- вывод оценки производительности поверх видеопотока (в FPS);
- оценка точности (accuracy) модели нейронной сети на тестовых данных, представленных в папке “test” на целевой машине;
- оценка “уверенности” алгоритма в решении задачи, при получении видеопотока от камеры.

## Конкурсное задание

Вариант конкурсного задания определяется с помощью жребия.

### **Задание 1. Распознавание эмоций человека в видеопотоке**

#### 1. Описание программного средства.

Разрабатываемое программное средство состоит из двух независимых частей:

Python-ноутбук (или Python-скрипт), в котором выполняется проектирование, обучение и/или дообучение модели нейронной сети для распознавания эмоций на основе входного изображения;

основное приложение, обеспечивающее взаимодействие с пользователем, написанное на Python с применением библиотеки PyQt5.

#### *1.1. Структура и наполнение Python-ноутбука (или Python-скрипта).*

В разрабатываемом скрипте должна быть представлена возможность указать 2 папки:

папка “train”, в которой размещаются изображения из тестового набора, на которой участник выполняет обучение и/или дообучение, а также тестирование модели нейронной сети;

папка “test”, которая будет использоваться для “слепого” тестирования членами комиссии и оценки точности на новых данных.

Используя рекомендуемые для предварительной установки библиотеки, участник должен реализовать базовую модель нейронной сети (допускается подключение и использование готовых моделей), оценить полученную точность (Accuracy).

Последующие эксперименты с применением аугментации и дообучения оформляются в виде отдельной модели нейронной сети.

Лучшая версия обученной модели нейронной сети должна сохраняться для последующего использования основным приложением. Название файла должно начинаться с фамилии участника, написанное латиницей.

При окончании обучения и формировании лучшей версии модели нейронной сети участники приглашают членов комиссии для демонстрации достигнутой точности (Accuracy), а также для организации “слепого” тестирования на изображениях в папке “test”.

Результаты работы ноутбука или скрипта:

вывод полученных метрик: точность (Accuracy) для тестовых наборов из папки 1 и папки 2, метрика F1;

построение матрицы ошибок (confusion matrix);

файл с сохраненной обученной моделью.

### *1.2. Организация работы основного приложения.*

При запуске основного приложения появляется главное меню, на котором пользователю предоставляется следующий функционал:

- элемент окна, в котором будет отображаться видеопоток от камеры;
- интерфейс выбора веб-камеры, настройки отображения информации с камеры;

- кнопка “запуск распознавания”;

- элемент окна, в котором будет отображаться дополнительная информация, определяемая самим участником;

- элемент окна, в котором предоставляется возможность выбрать подготовленную участником модель нейронной сети, а также при необходимости ее сконфигурировать.

Перед нажатием на кнопку “запуск распознавания” необходимо выбрать камеру и определить ее настройки.

При нажатии на кнопку “запуск распознавания”, программное средство должно выполнить:

- загрузку модели нейронной сети и подготовиться для ее использования;

- подключение камеры и организовать вывод видеопотока в соответствующий элемент окна.

Одновременно с этим начинается распознавание эмоций подготовленной участником моделью. По мере получения информации с результатами распознавания в окне поверх видеопотока отображается следующая информация:

- скорость обработки в кадрах в секунду (в FPS);

- “уверенность” модели в полученном результате.

## 2. Набор данных

Для разработки модели нейронной сети используется набор данных FER-2013 (<https://www.kaggle.com/datasets/msambare/fer2013>), представляющий собой коллекцию изображений для распознавания выражений лица, сгруппированных по 7 классам: злость (папка angry), отвращение (папка disgust), страх (папка fear), радость (папка happy), грусть (папка sad), удивление (папка surprise), нейтральное выражение (папка neutral). Количество изображений для каждого класса представлено в таблице 1. Каждое изображение представлено в оттенках серого и имеет размер 48x48 пикселей. Лица более или менее центрированы и занимают примерно одинаковое количество места на каждом изображении.

Таблица 1. Число изображений по классам

Название папки	Общее число файлов	
	папка “train”	папка “test”
angry	3995	958
disgust	436	111
fear	4097	1024
happy	7215	1774
neutral	4965	1233
sad	4830	1247
surprise	3171	831

## Задание 2. Поиск в видеопотоке объекта, задаваемого с помощью голосовых команд

### 1. Описание программного средства.

Разрабатываемое программное средство состоит из двух независимых частей:

Python-ноутбук (или Python-скрипт), в котором выполняется проектирование, обучение и/или дообучение модели нейронной сети для распознавания объектов на основе входного изображения;

основное приложение, обеспечивающее взаимодействие с пользователем, написанное на Python с применением библиотеки PyQt5.

#### *1.1. Структура и наполнение Python-ноутбука (или Python-скрипта).*

В разрабатываемом скрипте должна быть представлена возможность указать 1 папку:

папка “train”, в которой размещаются изображения из тестового набора, на которой участник выполняет обучение и/или дообучение, а также тестирование модели нейронной сети.

Используя рекомендуемые для предварительной установки библиотеки, участник должен реализовать базовую модель нейронной сети (допускается подключение и использование готовых моделей), оценить полученную точность (Accuracy).

Последующие эксперименты с применением аугментации и дообучения оформляются в виде отдельной модели нейронной сети.

Лучшая версия обученной модели нейронной сети должна сохраняться для последующего использования основным приложением. Название файла должно начинаться с фамилии участника, написанное латиницей.

При окончании обучения и формировании лучшей версии модели нейронной сети участники приглашают членов комиссии для демонстрации достигнутой точности (Accuracy).

Результаты работы ноутбука или скрипта:

вывод полученных метрик: точность (Accuracy) для тестовых наборов, метрика F1;  
 построение матрицы ошибок (confusion matrix);  
 файл с сохраненной обученной моделью.

### *1.2. Организация работы основного приложения.*

При запуске основного приложения появляется главное меню, на котором пользователю предоставляется следующий функционал:  
 элемент окна, в котором будет отображаться видеопоток от камеры;  
 интерфейс выбора веб-камеры, настройки отображения информации с камеры;  
 интерфейс выбора микрофона, его настройки;  
 кнопка “запуск распознавания”;  
 элемент окна, в котором будет отображаться дополнительная информация, определяемая самим участником;  
 элемент окна, в котором предоставляется возможность выбрать подготовленную участником модель нейронной сети, а также при необходимости ее сконфигурировать.

Перед нажатием на кнопку “запуск распознавания” необходимо выбрать камеру и микрофон, определить их настройки.

При нажатии на кнопку “запуск распознавания”, программное средство должно выполнить:

загрузку не менее одной модели нейронной сети и подготовиться для ее использования;

подключение камеры и организовать вывод видеопотока в соответствующий элемент окна.

Одновременно с этим начинается анализ голосовых команд, по мере распознавания которых выполняется поиск заданного объекта в видеопотоке. При обнаружении объекта, обвести его рамкой. По мере получения информации с результатами распознавания в окне поверх видеопотока отображается следующая информация:

скорость обработки в кадрах в секунду (в FPS);

скорость распознавания голосовых команд;

“уверенность” модели в полученном результате.

## 2. Набор данных

Набор данных определяется конкурсантом самостоятельно (могут использоваться, например, ImageNet, COCO). Программное средство должно поддерживать следующие классы (дополнительно приведен пример классов в базе ImageNet):

№	Название класса	ImageNet	
		ID класса	Название класса
1	Шариковая ручка	418	ballpoint, ballpoint pen, ballpen, Biro
2	Мобильный телефон	487	cellular telephone, cellular phone, cellphone, cell, mobile phone
3	Ноутбук	620	laptop, laptop computer
		681	notebook, notebook computer
4	Мячик для пинг-понга	722	ping-pong ball
5	Человек	-	-
6	Книга, записная книжка	-	-
7	Сумка, рюкзак	-	-

**Показатели и критерии оценок конкурса  
«Искусственный интеллект»**

№П/П	Критерий	Максимальное количество баллов
1	Дизайн-проект	20
2	Прототип	20
3	Оптимизация	40
4	Сборка и запуск проекта	20
Итого:		100

Участник, набравший наибольшее количество баллов, считается победителем.

В случае набора участниками одинакового количества баллов, победителем конкурса считается участник, набравший наибольшее

количество баллов и выполнивший конкурсное задание за наименьшее время.

В случае разногласий окончательное решение оценки конкурса принимает председатель жюри.