**V Международный конкурс исследовательских работ школьников «Research start» 2022/2023**

**Проектная работа**

**по теме:**

**«Нейросеть-классификатор мусора на изображениях»**

**Выполнил:**

**Ученик 9 «А» класса**

**школы МБОУ «Лицей «МОК №2»**

**Обухов Сергей Дмитриевич**

**Руководитель:**

**учитель информатики**

**Полякова Е.Г.**

**Воронеж-2023**

**Содержание**

[Введение 2](#_Toc125962458)

[Теоретическая часть 5](#_Toc125962459)

[Практическая часть 12](#_Toc125962460)

[Заключение 16](#_Toc125962461)

[Список используемой литературы 17](#_Toc125962462)

[Приложения 18](#_Toc125962463)

# **Введение**

Экологическая проблема – это изменение природной среды из-за антропогенных воздействий или стихийных бедствий, ведущее к нарушению структуры и функционирования природы. В современном мире существует множество экологических проблем, вызванных человеческой деятельностью. Большинство из них ещё не оказывают какого-либо заметного влияния на нашу жизнь. Но, безусловно, все экологические проблемы нуждаются в решении, ведь от этого зависит сама возможность существования человечества на Земле. Именно поэтому поиск способов решения различных экологических проблем стал одной из самых важных и актуальных проблем XXI века.

Многие экологические проблемы напрямую связаны с неправильной утилизацией огромного количества отходов. Так, например, множество мусорных свалок по всему миру вызывает загрязнение почвы, воды и воздуха; бездумная утилизация отходов заставляет производителей постоянно использовать новое сырьё, что приводит к нехватке природных ресурсов; разложение некоторых классов отходов на свалках влияет на изменение климата и здоровье людей и т.д. Эти и многие другие проблемы решает разделение мусора по классам и дальнейшая его переработка. Данный метод позволяет уменьшить загрязнение окружающей среды, сократить объём складируемого на свалках мусора, снизить потребление природных ресурсов за счёт использования переработанных отходов в качестве вторсырья и улучшить как экологическую, так и экономическую обстановку в мире в целом.

К сожалению, несмотря на неоспоримые плюсы разделения отходов, на данный момент сортировать и перерабатывать мусор должным образом получается только у небольшого количества самых развитых стран мира. Большинство же (в том числе и Россия) пока что только пытаются реализовать этот подход к утилизации. Например, в нашей стране перерабатывается менее 10% твёрдых бытовых отходов. Остальные 90% гниют на полигонах (суммарная площадь которых составляет около 4 миллионов гектаров), занимая свободное место и отравляя окружающую среду. Такая же ситуация наблюдается и во многих других странах. Это вызвано рядом причин, среди которых:

* Высокая стоимость (от 10 до 50 тысяч долларов) специального оборудования для сортировки мусора (гиперспектральных камер, грохотов, сепараторов и др.), что отталкивает предпринимателей и инвесторов от этой сферы;
* Нежелание людей разделять бытовой мусор самостоятельно вследствие неосведомлённости о пользе этого метода для человечества;
* Наличие огромного количества мусорных контейнеров без разделения отходов на классы, где пригодный для повторного использования мусор смешивается с непригодным и теряет возможность быть переработанным;
* Нежелание людей работать в плохих условиях на мусороперерабатывающих заводах, где на данный момент ручная сортировка отходов не имеет альтернатив.

Использование передовых технологий нейросетей и компьютерного зрения на мусороперерабатывающих предприятиях может почти полностью автоматизировать сортировку отходов, что приведёт к увеличению как качества, так и скорости этого процесса. Также применение нейросетей значительно снизит размер вложений, необходимых для содержания предприятия по сортировке мусора, за счёт практически полного отсутствия сотрудников и более дешёвого оборудования. Другой способ сортировки мусора с применением компьютерного зрения и нейросетей – это замена обычных мусорных баков на «умные» урны с автоматической сортировкой отходов по классам. Таким образом, мусор будет сортироваться ещё на этапе сбора и накопления отходов. Оба этих способа в будущем могут привести к увеличению доли сортируемых отходов вплоть до 100%, что положительно скажется на экологии и экономике. В этом и заключается актуальность моей проектной работы.

Также мой проект актуален из-за того, что в нём применяется очень актуальная в наши дни технология нейросетей. Уже сейчас существуют такие нейронные сети, способные лишить миллионы квалифицированных специалистов работы, ведь искусственный интеллект выполняет работу лучше и быстрее людей. Поэтому потенциал развития этой технологии в будущем практически безграничен.

Целью проекта является создание нейросети, способной определять тип представленного на изображении мусора (гофрокартон, электронные отходы, стекло, медицинские отходы, металл, бумага или пластик).

Для достижения цели мной поставлены следующие задачи:

1. Ознакомиться с информацией о сортировке мусора;
2. Изучить основные принципы работы нейронных сетей (в частности, свёрточных CNN);
3. Подготовить исходный набор данных с фотографиями различных типов мусора для обучения нейросети;
4. Разработать архитектуру нейросети;
5. Обучить модель нейросети на исходном наборе данных;
6. Опробовать нейросеть на практике и создать для неё пользовательский интерфейс.

# **Теоретическая часть**

Разделение мусора (или раздельный сбор мусора, селективный сбор мусора) – практика сбора и сортировки мусора в зависимости от его происхождения, состава и пригодности к переработке или вторичному использованию. Разделение мусора – это достаточно сложный многоступенчатый процесс. Его наиболее важным этапом является отделение пищевых отходов от остального мусора. Это позволяет избежать загрязнения перерабатываемого вторсырья и уменьшить объём свалочного газа на полигонах. Далее следует сортировка мусора на перерабатываемые и неперерабатываемые отходы. Очень часто применяется более глубокое разделение отходов – отдельно сортируются: стеклянная тара, многие виды пластика, макулатура, бытовой металлолом – подлежащий вторичной переработке мусор; опасные отходы (батарейки, аккумуляторы, медицинские отходы и т.д.), электронные отходы, некоторые виды бумаги (чеки, салфетки, тетрапак) и пластика (фольгированный, скотч, эколин) – неперерабатываемый мусор. В результате перерабатываемые отходы отделяются от неперерабатываемых и выделяются отдельные типы мусора, пригодные для вторичного использования.

Разделение мусора приносит много выгоды:

* Улучшение состояния окружающей среды: при сортировке мусора опасные отходы попадают на специализированные предприятия, где уничтожаются, не нанося вред природе;
* Экономия природных ресурсов: вторсырьё из переработанных (и ранее отсортированных) отходов используется для дальнейшего производства товаров. При этом остаются нетронутыми природные ресурсы, как, например, лес при производстве бумаги;
* Уменьшение количества мусора, хранящегося на свалках и полигонах: при разделении мусора увеличивается доля перерабатываемых отходов и сокращается количество мусора на свалках;
* Экономическая выгода: для предприятий выгодно применение в производственном цикле вторсырья, так как оно снижает себестоимость продукции; для государства выгодно использование вторсырья, так как благодаря ему экономятся невосполняемые природные ресурсы (такие как природный газ и нефть).

Человечество не задумывалось о разделении мусора до тех пор, пока в быту не получили широкое распространение синтетические материалы, что привело к росту количества неразлагающихся и токсичных отходов. Из-за этого в мире увеличилось количество мусорных свалок и полигонов. В то же время производства продолжали требовать новое сырьё. Во второй половине XX века страны Европы и Северной Америки столкнулись с мусорными кризисами. Для экономии природных ресурсов и уменьшения числа мусорных свалок твёрдые бытовые отходы стали вводиться во вторичный оборот. Наиболее ранний пример разделения отходов – сбор и повторное использование стеклотары, макулатуры и металлолома в некоторых странах (в том числе и в СССР) с первой половины XX века. Современная система сортировки отходов, при которой мусор разделяется на большее количество классов, появилась 1970-1980-е года. Первопроходцем в этой сфере стала ФРГ. В 1974 году в Западной Германии впервые появились отдельные мусорные баки для стекла, а в 1978 году – и для других классов (пластика, бумаги и т.д.). Примерно в те же годы первые опыты по сортировке мусора проводились в Японии. В 1990 году в Германии раздельный сбор мусора стал проводиться в общегосударственных масштабах. Вскоре так же поступили и в других странах Западной Европы. В 1990 году в Великобритании был издан закон о защите окружающей среды, в котором была закреплена необходимость разделения мусора. В 1992 году аналогичный закон был принят во Франции: он обязывал местные органы власти обеспечивать переработку мусора, что также привело к введению системы сортировки отходов. В 1990-е годы раздельный сбор и переработка мусора всё больше распространялись по всей Западной и Северной Европе, и в 2000-е разделение мусора стало в них практически повсеместным. Параллельно с Европой сортировка мусора развивалась и в Северной Америке. Например, в Канаде первые попытки по внедрению этого метода в повседневную жизнь осуществлялись ещё в начале 1980-х годов. Таким образом, к началу XXI века во всех развитых странах мира разделение мусора стало применяться практически повсеместно. На налаживание полностью функционирующей системы разделения мусора у них ушло от 20 до 30 лет. В настоящее время во многих европейских государствах (например, в Швеции, Германии и Швейцарии) благодаря развитой системе разделения и сортировки мусора перерабатывается до 90% всего мусора в стране. В Восточной Европе, в России и в других постсоветских странах этот подход к утилизации отходов получил распространение только в конце 2000-х – начале 2010-х годов. Именно поэтому в нашей стране разделение мусора находится только в зачаточной стадии. Тем не менее, Россия старается развиваться в этом направлении: в центрах многих городов стоят специальные баки для раздельного сбора отходов, функционируют предприятия по сортировке и переработке мусора, проводятся различные мероприятия, направленные на популяризацию разделения мусора среди россиян. В 2017 году в России стартовала реформа обращения с отходами производства и потребления («мусорная реформа»). Согласно ей к середине 2020-х годов планируется наладить во всей стране систему разделения и сортировки мусора, переработки вторичного сырья. Однако, по опросам ВЦИОМ, готовы к новому практике менее половины населения. Внедрение нейросетей в процесс сортировки мусора сделает его независимым от готовности и мнения граждан, что существенно ускорит развитие системы разделения отходов в России, а, следовательно, и увеличит объём перерабатываемого мусора.

Что же такое нейросети и что они умеют? Нейронная сеть – это последовательность нейронов, соединенных между собой синапсами (связями).

Начало развития нейронных сетей было положено в 1943 году, когда американские учёные Уорен Маккалок и Уолтер Питтс выпустили статью о том, как разработали компьютерную модель нейронной сети, опираясь на математические алгоритмы и теорию деятельности головного мозга. Уже тогда нейроны в их модели были способны обучаться путём подстройки параметров, описывающих синаптическую проводимость между ними. Исследователи предположили, что такая сеть в будущем станет в состоянии распознавать образы, обобщать, т. е. обладает всеми чертами интеллекта, что оказалось правдой. В 1958 году американский учёный Фрэнк Розенблатт изобрёл первый однослойный перцептрон – примитивную математическую модель обработки информации человеческим мозгом. В 1960 году перцептрон Розенблатта смог решить простейшую задачу классификации – распознавание отдельных символов на карточках. Далее, вплоть до XXI века, исследования в области нейросетей на время прекратились из-за недостаточных мощностей компьютеров того времени. В 2007 году был сделан важнейший шаг в сфере разработки нейронных сетей – британский математик Джеффри Хинтон создал алгоритмы глубокого обучения многослойных нейронных сетей, что открыло перед человечеством практически безграничные возможности в сфере нейросетей и искусственного интеллекта. Особенно быстрыми темпами эта сфера стала развиваться в последние два года. Появились такие нейросети, как Midjourney и DALL-E, создающие неповторимые живописные картины по одному лишь текстовому описанию, генераторы текстов GPT-3 и Балабоба, способные написать целые литературные произведения, ChatGPT, способный писать программный код для решения простых задач программирования, и многие другие.

Сейчас существует бесчисленное множество нейросетей, выполняющих самые разные задачи от распознавания лиц на изображениях до предсказания стоимости криптовалют на бирже. Все эти задачи условно разделяют на 3 категории: классификация (распознавание образов, голосов, постановка диагнозов), предсказание (прогноз погоды, стоимости ценных бумаг и т.д.), моделирование (поведение системы в определённых условиях). Поэтому фундаментальные алгоритмы, применяемые во всех нейросетях, в целом, примерно схожи между собой.

Все нейросети состоят из нейронов – вычислительных единиц, которые получают информацию, производят над ней простые вычисления и передают её дальше по нейросети. Нейроны образуют слои – совокупности нейронов. Выделяют 3 основных вида слоёв в нейросетях: входной (получает исходную информацию), несколько скрытых (обрабатывают исходную информацию) и выходной (выводит результат работы нейросети). Каждый нейрон связан с определёнными нейронами (или всеми) из предыдущего слоя с помощью синапсов, обладающих некоторыми весами. Именно благодаря этим весам входная информация изменяется, когда передается от одного нейрона к другому. Только во входном слое входная информация в нейронах этого слоя равна выходной. Во всех остальных слоях входная информация в каждом нейроне – это взвешенная сумма информации из каждого нейрона, связанного с данным. Другими словами, входная информация в каждый нейрон – это сумма всех входных данных, умноженных на соответствующие им веса. После того, как нейрон получил входную информацию, он совершает над ней некоторые простые вычисления. Под простыми вычислениями подразумевается нормализация (приведение к определённому диапазону значений) полученной взвешенной суммы с помощью специальной функции активации и смещения нейрона. Смещение нейрона – это просто число, которое добавляется к взвешенной сумме. Функция активации (например, сигмоида, ReLU и др.) представляет собой нелинейное преобразование над взвешенной суммой, благодаря которому информация в нейронах после нескольких слоёв не становится слишком большой из-за каскадного эффекта. В результате выходная информация в нейронах принимает значения в диапазонах [-1,1] или [0,1] и передаётся дальше по нейросети. Эти операции повторяются до выходного слоя, который выдаёт конечный результат работы нейросети. Такой проход информации по нейронам от входного слоя к выходному называется прямым распространением. В принципе, можно сказать, что нейросеть – это просто массивная функция с миллионами различных параметров (весов, смещений), которая получает на вход какие-то данные, совершает над ними различные преобразования и выдаёт конечный результат. Но чтобы нейросеть работала корректно, необходимо подобрать и настроить все эти параметры под решение конкретной задачи. Эта «настройка» параметров называется обучением нейросети. Обучение имеет несколько характеристик:

1. Тренировочный размеченный датасет – набор данных, на которых обучается нейросеть, в котором каждому примеру соответствует некоторый правильный ответ.
2. Коэффициент скорости обучения – характеристика того, насколько сильно и быстро будут изменяться параметры нейросети в процессе обучения. Чем он выше, тем быстрее пройдёт обучение, но при этом понизиться конечная точность нейросети. И наоборот, чем он ниже, тем медленнее пройдёт обучение, но при этом повысится конечная точность нейросети.
3. Эпоха – это большой обучающий цикл, состоящий из массивного набора входных данных. Эпоха увеличивается каждый раз, когда через нейросеть проходит весь набор тренировочных датасетов.
4. Итерация – это общее количество тренировочных датасетов, прошедших через нейронную сеть.
5. Батчи – это небольшие пакеты данных одной итерации. Исходный тренировочный датасет перед обучением нейросети разбивается на батчи установленного размера. В процессе обучения нейросети на вход подаются эти сформированные батчи данных.
6. Функция ошибки (кросс-энтропия, Mean Squared Error и др.) – функция, отражающая расхождение между ожидаемыми правильными ответами из тренировочного датасета и результатами работы нейросети.
7. Точность нейросети – отношение количества правильных ответов, данных нейросетью, к общему числу данных ответов. Эта величина обратно пропорциональна значению функции ошибки (чем больше значение функции ошибки, тем меньше точность, и наоборот).

 Процесс обучения нейросети начинается с уже описанного ранее прямого распространения, в результате которого нейросеть по определённым входным данным из тренировочного датасета выдаёт какой-то результат. Далее этот результат сравнивается с ожидаемыми правильными ответами из тренировочного датасета, и на основе этого вычисляется так называемая функция потерь (или функция ошибки). Основываясь на значении этой функции, начинается процесс обратного распространения ошибки. Его суть заключается в последовательном изменении параметров нейросети (весов и смещений), начиная с выходного слоя. Изменения происходят таким образом, чтобы значение функции потерь уменьшалось, а точность нейросети увеличивалась. Таким образом, главная задача обучения – подобрать такие значения параметров нейросети, чтобы значение функции потерь было минимально. Оптимальные значения весов и смещений подбираются с помощью некоторых очень сложных алгоритмов, самым популярным из которых является метод градиентного спуска. Все эти действия повторяются снова и снова, пока нейросеть полностью не обучится и не станет выдавать нужные результаты.

Задача моей нейросети – классификация изображений по типам мусора на них. Поэтому свою нейросеть я буду разрабатывать на основе свёрточных CNN, ведь именно этот тип нейросетей показывает наилучшие результаты на задачах классификации и распознавания изображений. Главное отличие CNN от других нейросетей – наличие операции свёртки в процессе обработки входного изображения. Благодаря этому алгоритму выделяются основные компоненты и отсеиваются шумы на изображении, а также понижается объём данных, что существенно ускоряет процесс обучения таких нейросетей.

# **Практическая часть**

Теперь, рассмотрев теоретические основы проекта, перейдём непосредственно к созданию нейронной сети.

Первым, но, пожалуй, самым важным этапом разработки нейросети является подготовка тренировочного датасета, ведь качество работы нейросети прямо пропорционально качеству и количеству данных в датасете. Для своего проекта я объединил сразу 2 датасета с фотографиями мусора из свободного доступа – Trashnet и TrashBox (оба найдены на GitHub). После удаления некоторого количества повреждённых и неправильных данных (в основном из TrashBox) я получил свой собственный датасет с 7 категориями: гофрокартон (498 изображений), электронные отходы (499 изображений), стекло (501 изображение), медицинские отходы (483 изображения), металл (502 изображения), бумага (500 изображений) и пластик (500 изображений). Всего в датасете оказалось 3 483 изображения на 250 МБ, что является довольно небольшим размером для датасета.

Далее началась программная часть разработки нейросети. Для обучения нейросети я использовал язык программирования Python, библиотеку PyTorch и среду разработки Jupyter Notebook – словом, все лучшие на данный момент инструменты для работы с нейросетями.

После загрузки датасета на сервер Jupyter над ним были произведены следующие манипуляции:

1. Преобразование всех изображений из датасета к трёхмерным тензорам $256×256×3$;
2. Разделение исходного датасета на тренировочный (79%), проверочный или валидационный (7%) и тестовый (14%) наборы данных;
3. Разбиение тренировочного и проверочного датасетов на батчи по 32 и 64 фотографии соответственно;
4. Перенос вычислений из ЦП в GPU с помощью специальной технологии параллельных вычислений Nvidia CUDA. Это существенно уменьшит время обучения нейросети, ведь видеокарты гораздо более приспособлены к работе с тензорами, чем центральные процессоры.

Далее была разработана архитектура нейросети. За основу я взял свёрточную нейросеть ResNet50. В итоге, моя нейросеть обладает следующими характеристиками:

* Входной слой – $256×256×3$ нейронов – количество пикселей на каждом изображении, умноженное на количество каналов изображения (3 канала, так как формат – RGB);
* Скрытые слои – 4 слоя ResNet50 с реализованными в них некоторыми алгоритмами обработки входных изображений (в том числе и со встроенной функцией свёртки);
* Выходной слой – 7 нейронов – количество типов мусора. В них находятся значения, характеризующие уверенность нейросети в том, что на фотографии представлен соответствующий тип мусора. Следовательно ответ нейросети – это номер нейрона выходного слоя с наибольшим выходным значением;
* Изначально параметры нейросети заполнены не случайными числами, а весами и смещениями из предобученной ResNet50. Это значит, что нейросеть уже умеет определять самые простейшие формы и фигуры на изображениях, что существенно упростит процесс обучения;
* Используемая функция потерь – кросс-энтропия;
* Используемая функция активации – сигмоида;
* Коэффицент скорости обучения = 0.000055 (достаточно медленное, но зато качественное обучение);
* Количество эпох = количество итераций (т.к. я использовал единственный датасет в обучении) = 6;

Такая нейросеть до какого-либо обучения уже показывает относительно неплохие результаты на валидационном наборе данных (из-за параметров от предобученной ResNet50): точность = 13%, а значение функции потерь = 1,9455.

Далее переходим ко второму самому важному этапу разработки нейросети – к её непосредственному обучению. На первой же эпохе точность нейросети очень сильно возросла – от 13 до 88%. А далее, вплоть до шестой эпохи, нейросеть понемногу улучшала свои показатели. Нейросеть обучалась примерно 15 минут. В результате, конечные показатели нейросети оказались следующими: точность = 95%, значение функции потерь = 1,22. Это означает, что обучение прошло как надо, ведь значение функции потерь уменьшилось, а точность нейросети, наоборот, повысилась по сравнению с первоначальными значениями.

При тестировании нейросети на изображениях из тестового набора данных, которые она ещё никогда не видела, доля правильных ответов составила более 93% (453 правильно классифицированных изображения из 487). При этом средняя уверенность нейросети в ответах оказалась равна 97,8%.

После этого я провёл многочисленные тесты работы нейросети на фотографиях, сделанных мной. На большинстве фотографий она выдавала правильные ответы, но всё-таки иногда встречались ошибки. Они связаны в первую очередь с малым количеством обучающих данных. Для создания нейросети, способной безошибочно определять тип мусора на изображениях, нужны гораздо большие по размеру датасеты с более чем 50 тысячами фотографий мусора.

Для удобного пользования нейросетью я создал для неё компьютерное приложение с графическим пользовательским интерфейсом с помощью GUI-фреймворка PySimpleGUI и редактора кода Visual Studio Code. Для этого пришлось также сохранить все установленные в процессе обучения нейросети веса и смещения в специальном файле с расширением .pth. С его помощью можно интегрировать мою нейросеть в другие проекты – достаточно просто создать новую модель нейросети с такими же характеристиками, как и разработанная, а затем загрузить в неё параметры из вышеупомянутого файла. В результате получится полностью корректно функционирующая нейросеть-классификатор мусора на изображениях.

Последним пунктом в процессе разработки нейросети стало преобразование исходного кода приложения к исполняемому файлу .exe с помощью утилиты PyInstaller. Теперь моё приложение со встроенной нейросетью сможет запуститься на абсолютно любом компьютере, и теперь каждый желающий может опробовать его на практике.

# **Заключение**

В ходе выполнения этой проектной работы я рассмотрел один из самых эффективных способов решения множества экологических и экономических проблем современного мира – разделение мусора. Я ознакомился с существующими методами реализации этого подхода к утилизации отходов в России и за рубежом, оценил их плюсы и минусы и нашёл, как мне кажется, наиболее инновационный и перспективный – с применением технологии нейросетей.

Я узнал много нового о том, как устроены и как работают нейросети, познакомился с библиотекой PyTorch, средой разработки Jupyter Notebook и технологией параллельных вычислений Nvidia CUDA, тем самым получив много полезного для моего будущего опыта, улучшил свои умения по работе с Python и PySimpleGUI, а самое главное – добился поставленной цели и создал нейросеть, способную определять тип представленного на изображении мусора.

Я надеюсь, что не остановлюсь на достигнутом и всячески улучшу нейросеть, добавив в неё новые классы мусора, такие как текстиль, резина и другие, минимизирую количество ошибочных ответов, а также смогу интегрировать свою разработку на какое-либо предприятие по сортировке мусора или в «умные» урны.

# **Список используемой литературы**

1. Алексеенко В. Б., Сопилко Н. Ю., Балалаев А. К., Лисицкая С. М. Эколого-экономическое обоснование применения раздельного сбора ТБО.
2. Горбань А.Н. Обучение нейронных сетей.
3. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение.
4. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика.
5. Преликова Е. А., Юшин В. В., Вертакова Ю. В. Эколого-экономические приоритеты раздельного сбора отходов.
6. Путинцева Н. А. Организация раздельного сбора твердых коммунальных отходов в России.
7. Тадеусевич Рышард, Боровик Барбара, Гончаж Томаш, Леппер Бартош. Элементарное введение в технологию нейронных сетей с примерами программ.
8. Ферцев А. А. Реализация нейронной сети для распознавания изображений с помощью технологии Nvidia CUDA.
9. Хайкин С. Нейронные сети: полный курс.
10. Ясницкий Л. Н. Введение в искусственный интеллект.

# **Приложения**

1. Исходный код обучения нейросети представлен в соответствующем pdf файле.
2. Исходный код приложения с интегрированной нейросетью:

import torch

import torchvision.models as models

import torch.nn as nn

import torchvision.transforms as transforms

from PIL import Image

import PySimpleGUI as sg

import os

import io

rus\_kinds = ["Гофрокартон","Электронные отходы","Стекло","Медицинские отходы","Металл","Бумага","Пластик"]

transformation = transforms.Compose([transforms.Resize((256,256)), transforms.ToTensor()])

model\_path = "Rubbish Classificator AI.pth"

file\_types = [("JPEG (\*.jpg)", "\*.jpg"),("All files (\*.\*)", "\*.\*")]

def accuracy(outputs, answers):

 \_, predictions = torch.max(outputs,1)

 correct = torch.sum(predictions == answers)

 acc = correct/len(answers)

 return acc

class GarbageClassificator(nn.Module):

 def \_\_init\_\_(self):

 super().\_\_init\_\_()

 self.network = models.resnet50(weights="DEFAULT")

 n=self.network.fc.in\_features

 self.network.fc = nn.Linear(n,7)

 def predict(self,image):

 image = transformation(image)

 img = image.unsqueeze(0)

 preds = self(img)

 \_, ans = torch.max(preds,1)

 return preds,ans

 def forward(self,batch):

 res = torch.sigmoid(self.network(batch))

 return res

def main():

 rubbish\_classificator = GarbageClassificator()

 rubbish\_classificator.load\_state\_dict(torch.load(model\_path,map\_location=torch.device("cpu")))

 rubbish\_classificator.eval()

 font\_norm = ("Helvetica",12)

 font\_big = ("Arial",16)

 sg.theme("BlueMono")

 layout\_main = [[sg.Text("Загрузить фотографию:"),sg.Input(key="directory",disabled=True),sg.FileBrowse("Искать",key="image\_browser",file\_types=file\_types),sg.Button("Определить", key = "call\_ans")],

 [sg.Push(),sg.Image(key="image"),sg.Push()],

 [sg.Text("Тип мусора на изображении:",font=font\_big,text\_color='red'),sg.Text("",key="ans",font=font\_big,text\_color='red')],

 [sg.Text("")],

 [sg.Text("Гофрокартон:",font=font\_norm),sg.Text("0%",key = "Гофрокартон",font=font\_norm)],

 [sg.Text("Электронные отходы:",font=font\_norm),sg.Text("0%",key = "Электронные отходы",font=font\_norm)],

 [sg.Text("Стекло:",font=font\_norm),sg.Text("0%",key = "Стекло",font=font\_norm)],

 [sg.Text("Медицинские отходы:",font=font\_norm),sg.Text("0%",key = "Медицинские отходы",font=font\_norm)],

 [sg.Text("Металл:",font=font\_norm),sg.Text("0%",key = "Металл",font=font\_norm)],

 [sg.Text("Бумага:",font=font\_norm),sg.Text("0%",key = "Бумага",font=font\_norm)],

 [sg.Text("Пластик:",font=font\_norm),sg.Text("0%",key = "Пластик",font=font\_norm)]]

 window\_main = sg.Window("Rubbish Classificator",layout\_main)

 while True:

 event,values=window\_main.read()

 if event==sg.WIN\_CLOSED:

 break

 if event=="call\_ans":

 for i in range(7):

 window\_main[rus\_kinds[i]].update(text\_color='black')

 image\_path = values["directory"]

 if os.path.exists(image\_path):

 image = Image.open(image\_path)

 preds, ans = rubbish\_classificator.predict(image)

 kind = rus\_kinds[ans]

 window\_main["ans"].update(kind)

 for i in range(7):

 act\_conf = round(preds[0][i].item()\*100,3)

 window\_main[rus\_kinds[i]].update(str(act\_conf)+"%")

 window\_main[kind].update(text\_color='red')

 image.thumbnail((400, 400))

 image\_bytes = io.BytesIO()

 image.save(image\_bytes, format="PNG")

 window\_main["image"].update(image\_bytes.getvalue())

 window\_main.close()

if \_\_name\_\_=="\_\_main\_\_":

 main()