

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

**Проектирование и разработка методов интеграции моделей ИИ в  
системы, создание дизайнерских решений к ним**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Черняковой Дарьи Владиславовны

Научный руководитель  
доцент, к.ф.-м.н., доцент

О.С. Кузнецова

Консультант  
доцент, к.ф.-м.н., доцент

Д. В. Мельничук

Зав. кафедрой  
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2025

**Введение.** Стартап «Помощник ИИ по уходу за растениями» предлагает эффективное решение для любителей растений и профессиональных садоводов — чат-бот, использующий искусственный интеллект для идентификации растений, диагностики заболеваний и предоставления рекомендаций по уходу. Проект направлен на улучшение здоровья растений и повышение уровня знаний пользователей о растениях на фоне роста интереса к экологии. Основная цель проекта – удовлетворить потребность пользователя в быстрой и эффективной информации касательно ухода за растениями.

Актуальность проекта обосновывается тем, что около 80% хозяйств России имеют комнатные растения. К тому же, на фоне интереса к спасению растений от рынка быстрого потребления, продукт может быть актуален за пределами России. Отечественных конкурентов найдено не было, чтобы конкурировать с зарубежными приложениями, упор будет делаться на стиль и дополнительные функции. А именно: пиксель-арт и оформление в виде визуальной новеллы, из дополнительных функций разговор с персонажем, интерактивный элемент в виде генерации реалистичных изображений на тему природы. Есть возможность быстро добавлять и убирать отдельные функции, структура проекта очень гибкая, потому что инновационность достигается за счет использования современных технологий, к примеру, сверточной нейронной сети и генеративно-состязательной сети. Также актуальность для пользователя повышается за счет сегментации рынков, отсутствия отечественных аналогов, визуально выделяющегося способа предоставления информации. Все это нужно, чтобы с большей вероятностью привлечь к проекту молодых людей, которые и сами не против использовать растения для создания базы для контента вокруг своей онлайн-персоны.

Актуальность самой работы состоит в том, что в практической части подробно разобрана интерактивная часть системы - GAN или Generative adversarial network, генеративно-состязательная сеть. То есть работа актуальна за счет относительной новизны инструмента, а также за счет подхода к объяснению и использованию. Текст работы может быть использован как руководство к интеграции современных решений в другие проекты, поскольку распространенной практикой является использование ИИ моделей или же

модели искусственного интеллекта именно так, как описано в основном тексте, то есть через API – application programming interface.

Целью данной работы является создание чат-бота на основе моделей ИИ. Основная функция бота – умение распознавать растение по фото и рекомендовать уход для него. Чат-бот должен уметь общаться с пользователем на отвлечённые темы в рамках прописанной истории, используя имеющийся в коде промпт.

Задачи данной работы:

1. Проанализировать информацию, чтобы можно было обосновать выбранную идею стартапа.
2. Создать макет системы с точки зрения дизайна и бэкенда.
3. Дать описание основных проблем, ради решения которых будет создана система.
4. Решить поставленные системе задачи.
5. Показать гибкость системы за счет добавления дополнительных функций.

**Первый раздел** призван описать проект в общем, включая обоснование идеи, описание различных подготовительных мероприятий, необходимых на конечном этапе проекта, но имеющих смысл применения уже на этапе разработки. Например, текст для патента проекта.

Первый пункт раздела описывает текущее состояние рынка домашних растений, различные характеристики возможных потребителей, общее описание идеи с обоснованием на основе сделанных в этом разделе выводов.

Анализ рынка — это исследование текущей ситуации в определённой отрасли с целью выявления трендов, уровня конкуренции, потребностей клиентов и других факторов, влияющих на бизнес. Этот процесс помогает принимать обоснованные решения, минимизировать риски и находить лучшие стратегии для развития своего проекта. Процесс включает в себя сбор, систематизацию и интерпретацию информации о различных аспектах рынка. Это очень важный этап, но поскольку идея была выбрана удачно, анализ не занял много времени. Например, согласно опросу SWNS – британская медиакомпания, 75% молодых людей в возрасте от 25 до 39 лет покупают растения, чтобы увидеть, достаточно ли они ответственны, чтобы завести настоящего

питомца. В России же около 80% хозяйств ухаживают за домашними растениями. К такому же выводу приходят и крупные компании, которые используют горшочные растения как декор, чтобы продать свой товар лучше. Если брать более конкретный пример, то продать недвижимость можно успешнее, если поставить пару растений в помещении. То есть растения влияют на человека психологически в самых разных областях. Второй пункт объясняет необходимость патента как способа, формулу продукта для данного проекта.

Таким образом, домашние растения являются неотъемлемой частью современной жизни человека. Поэтому идея чат-бота, который будет помогать в уходе за растениями, небезосновательна и жизнеспособна.

В третьем пункте есть информация о структуре системы с точки зрения дизайна и кода на этапе планирования, также рассказано про отличия от последующих версий. Описание структуры системы, полученное в процессе применения различных подходов к описанию системы было необходимо, чтобы наладить командную работу и в будущем связать все элементы вместе.

Система состоит из трех моделей ИИ: Mistral, styleGAN3 и Plant.id с точностью распознавания растений, равной 93%. Чтобы рядовой пользователь мог их использовать, была введена визуальная составляющая в виде веб-страниц с интерфейсом.

Изначальное решение предполагало разделение чата с рекомендациями и чата с персонажем – Розой. Этот вариант был отвергнут, поскольку интерфейсы чатов совпадали и могли ввести пользователя в заблуждение. Также, после исследования рынка моделей, было принято решение отказаться от модели Character.ai, которую теперь заменяет Mistral. Все потому что управляющие этого проекта принимали решения, сильно ограничивающие работу модели. Но замена произошла быстро благодаря легко расширяемой структуре и благодаря блочности кода.

Есть страница с информацией о проекте и регистрацией, к которой с помощью бэкэнда прикреплена база данных. Есть главная страница выбора, где пользователь может нажать на одну из двух кнопок и перейти либо на страницу с чатом, либо на страницу с интерактивным ивентом. На странице с чатом пользователь может отправить фото или просто попросить рассказать

про уход конкретного растения, если он знает название. Есть два варианта развития событий:

1. Если это было изображение, то оно отправится Plant.id, тогда модель вернет класс растения, то есть его название. Также есть возможность сделать так, чтобы модель оценивала состояние здоровья растения. Все это отправляется модели Mistral без каких-либо дополнительных действий пользователя.
2. Если пользователь знает название, то промпт отправляется Mistral, которая возвращает ответ.

Далее, если пользователь явно не покажет намерения узнать уход, то есть если он не отправит фото или название, то бот будет переписываться с пользователем в рамках прописанных правил.

Третья модель нужна для обеспечения интерактивной составляющей, к которой пользователи могут перейти с прошлой главной страницы с чатами. Такое решение было принято, чтобы привлечь больше людей, показать гибкость проекта вероятным инвесторам. При необходимости интерактивное событие можно будет убрать из основной версии проекта, как и любую другую модель.

Также кратко рассказано про выбранный стек технологий, включающий в себя следующие основные инструменты: Python, PyTorch, VS 2019 с соответствующими библиотеками для C++ разработки, VS Code как IDE, Flask.

В четвертом подразделе рассказывается в целом об идее, делаются выводы о ее реализуемости. Также рассказывается о преимуществах идеи, к примеру, о выбранной стилистике проекта:



Рисунок 1 — Страница выбора

На данный момент дизайн меняется в соответствии с современными запросами пользователей, поэтому конечный вид продукта будет немного другим.

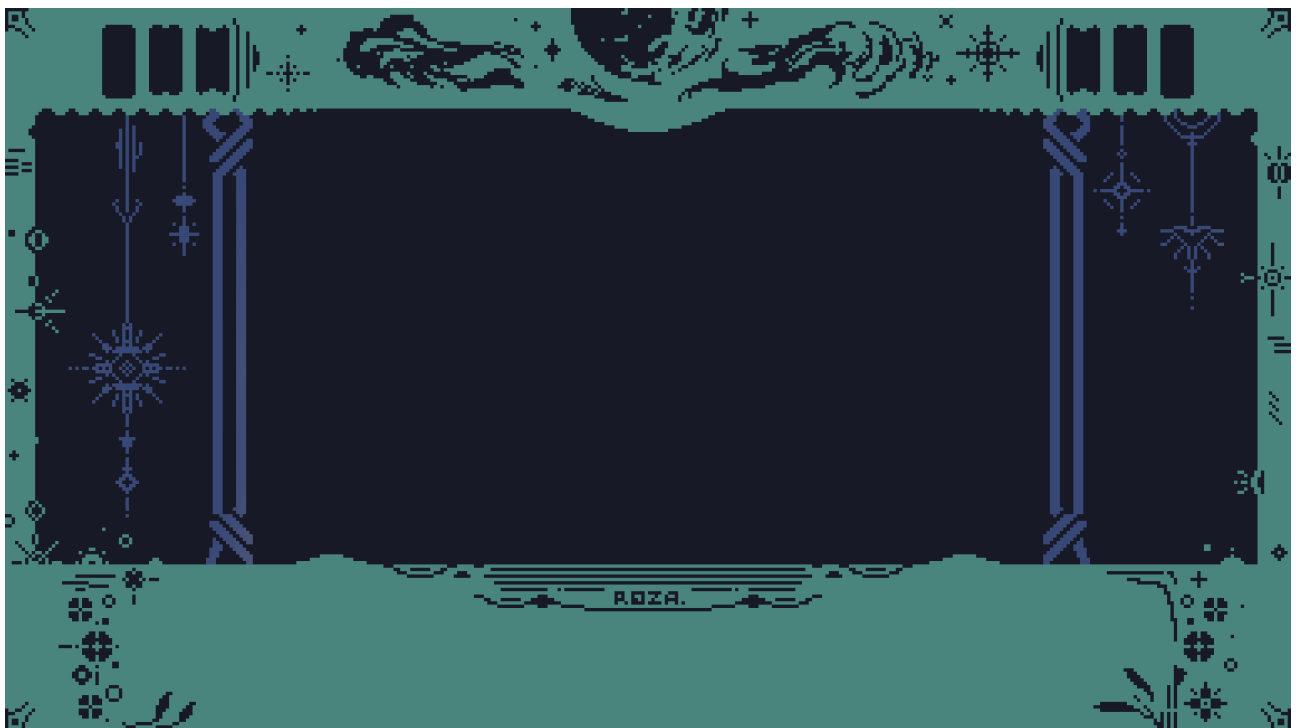


Рисунок 2 — Часть нового дизайна чата

**Во втором разделе** более подробно рассказывается об основных решаемых системой проблемах, а именно о задаче классификации изображений, об имитации моделью человеческой письменной речи, то есть о задаче NLP, также о задаче генерации изображений.

Подробно рассказывается про структуру различных нейронных сетей, ключевые моменты и идеи при создании моделей. Это важно, чтобы на этапе разработки понимать, как предобработать информацию от пользователя, чтобы модель более эффективно находила решение. Так, к любому сообщению пользователя в начале всегда добавляется инструкция, чтобы ответ приходил на русском языке.

**Третий раздел** является практической частью работы. В нем рассмотрены практические решения поставленных задач. Ниже предоставлен код для обработки ответа от модели Plant.id моделью Mistral.

```
1 def ChatMistral(message, image, user_id):
2     if(len(image) == 0):
3         print('No image')
4         instruction = """
5         Говори только на русском.
6         Давайте поиграем в ролевую игру.
7         Всегда давайте один ответ.
8         Только ролевая игра, только диалог.
9         Не отправляй никаких комментариев.
10        Не отправляй никаких заметок.
11        Всегда помогай с заботой, уважением и говори правду.
12        Отвечай максимально полезно, но безопасно.
13        Избегай вредоносного, неэтичного, предвзятого
14        или негативного контента.
15        Убедись, что ответы способствуют справедливости
16        и позитиву.
17
18        Тебя зовут Роза. Обладаешь
19        спокойным характером, задумчива...
20        """
```

```

21     chat_response = client.chat.complete(
22         model=MODEL,
23         messages=[
24             {"role": "system", "content": instruction},
25             {"role": "user", "content": message},
26
27         ],
28     )
29 else:
30     print('With image')
31     plantClass = classify_plant(image, user_id)
32     type(plantClass)
33     disease = classify_disease(image, user_id)
34     if disease=='None':
35         chat_response = client.chat.complete(
36             model = MODEL,
37             messages = [
38                 {
39                     "role": "user",
40                     "content": "Предоставь ответ на
41 следующий вопрос на русском: "
42 + message + plantClass
43 + ", как ухаживать за ним?",
44                 },
45             ]
46         )
47     else:
48         chat_response = client.chat.complete(
49             model = MODEL,
50             messages = [
51                 {
52                     "role": "user",
53                     "content": "Предоставь ответ на

```



```

54         следующий вопрос на русском: "
55         + message + plantClass +
56         ", как ухаживать за ним?" +
57         "Скажи, что растение, возможно,
58         болеет и предоставь название
59         болезни на русском:" + disease,
60     },
61 ]
62 )
63 print(chat_response.choices[0].message.content)
64 return chat_response.choices[0].message.content

```

В данном разделе подробно рассмотрен процесс донастройки и дообучения готовой GAN модели на предобработанном датасете. Какова задача GAN в данной работе? Модель привносит в проект интерактивный элемент, напрямую связанный с одной из основных тем проекта – с природой. Эта часть проекта не обязательна, но призвана показать вполне конкретный уровень расширяемости проекта. Так, если в будущем будет принято решение сменить направленность работы на сельское хозяйство, то можно будет легко добавить вместо GAN иную модель, к примеру, для определения типа почвы.

Датасет обрабатывался с помощью другого GAN, который помог узнать, какие объекты есть на изображениях и отсортировать изображения с ненужными классами. Ниже представлен код основной функции:

```

1  def process(crf, images):
2
3      postprocessor = setup_postprocessor(CONFIG) if crf else None
4
5      model.load_state_dict(state_dict)
6      model.eval()
7      model.to(device)
8
9      print("Image:", images)
10     image = cv2.imread(images, cv2.IMREAD_COLOR)

```

```

11     image, raw_image = preprocessing(image, device, CONFIG)
12     labelmap = inference(model, image, raw_image, postprocessor)
13     labels = np.unique(labelmap)
14     colors = {
15         0: [255, 0, 0],
16         1: [0, 255, 0],
17         2: [0, 0, 255],
18         3: [255, 255, 0],
19         4: [255, 0, 255],
20         ....
21     }
22
23     labels = np.unique(labelmap).astype(int)
24     used_colors_set = set()
25     combined_mask = np.zeros((raw_image.shape[0],
26     raw_image.shape[1], 3), dtype=np.uint8)
27     for label in labels:
28         mask = (labelmap == label)
29         if label in colors:
30             specified_color = colors[label]
31             combined_mask[mask] = specified_color
32     if not used_colors_set.intersection(forbidden):
33         output_path = "segmentated\\" +
34         str(images[6:12]) + ".png"
35         cv2.imwrite(output_path, image)
36     else:
37         print(f"Изображение {images}
38         содержит запрещенные классы и не будет сохранено.")
39     return(used_colors_set)

```

Далее был запущен процесс обучения модели, который можно увидеть на рисунке ниже 3.

```

1 loading training set...
2
3 Num Images: 2372
4 Image shapes: [0, 512, 512]
5 Label shape: [0]
6
7 Constructing networks...
8 Resuming from "D:\out\00001-stylegan3-r-cropped512-gpus1-batch8-gamma6.6\network-snapshot-000160.pkl"
9 Setting up PyTorch plugin "bias_act_plugin"... Done.
10 Setting up PyTorch plugin "filtered_lrelu_plugin"... Done.
11
12 Generator      Parameters  Buffers  Output shape  Datatype
13 ---
14 mapping_fc8    262656    -         [4, 512]      float32
15 mapping_fc1    262656    -         [4, 512]      float32
16 mapping        -         512       [4, 16, 512]  float32
17 synthesis.input.affine  2852     -         [4, 4]        float32
18 synthesis.input  1048576   3081      [4, 1024, 36, 36] float32
19 synthesis.L0_36_1024.affine  525312   -         [4, 1024]     float32
20 synthesis.L0_36_1024  1049600   157       [4, 1024, 36, 36] float32
21 synthesis.L1_36_1024.affine  525312   -         [4, 1024]     float32
22 synthesis.L1_36_1024  1049600   157       [4, 1024, 36, 36] float32
23 synthesis.L2_32_1024.affine  525312   -         [4, 1024]     float32
24 synthesis.L2_32_1024  1049600   169       [4, 1024, 52, 52] float32
25 synthesis.L3_52_1024.affine  525312   -         [4, 1024]     float32
26 synthesis.L3_52_1024  1049600   157       [4, 1024, 52, 52] float32
27 synthesis.L4_84_1024.affine  525312   -         [4, 1024]     float32
28 synthesis.L4_84_1024  1049600   169       [4, 1024, 84, 84] float16
29 synthesis.L5_84_1024.affine  525312   -         [4, 1024]     float32
30 synthesis.L5_84_1024  1049600   157       [4, 1024, 84, 84] float16
31 synthesis.L6_148_1024.affine  525312   -         [4, 1024]     float32
32 synthesis.L6_148_1024  1049600   169       [4, 1024, 148, 148] float16
33 synthesis.L7_148_967.affine  525312   -         [4, 1024]     float32
34 synthesis.L7_148_967  991175    157       [4, 967, 148, 148] float16
35 synthesis.L8_276_645.affine  496071   -         [4, 967]      float32
36 synthesis.L8_276_645  624368    169       [4, 645, 276, 276] float16
37 synthesis.L9_276_431.affine  338885   -         [4, 645]      float32
38 synthesis.L9_276_431  278425    157       [4, 431, 276, 276] float16
39 synthesis.L10_532_287.affine  221103   -         [4, 431]      float32
40 synthesis.L10_532_287  123984    169       [4, 287, 532, 532] float16
41 synthesis.L11_532_192.affine  147231   -         [4, 287]      float32
42 synthesis.L11_532_192  52596     157       [4, 192, 532, 532] float16
43 synthesis.L12_532_128.affine  80496    -         [4, 192]      float32
44 synthesis.L12_532_128  24704     25        [4, 128, 532, 532] float16
45 synthesis.L13_512_128.affine  65664    -         [4, 128]      float32
46 synthesis.L13_512_128  16512     25        [4, 128, 512, 512] float16
47 synthesis.L14_512_3.affine  65664    -         [4, 128]      float32
48 synthesis.L14_512_3    387       1         [4, 3, 512, 512]  float16

```

Рисунок 3 — log файл, показывающий процесс обучения модели

Ниже представлены последние значения различных метрик, которые записываются в json файл в процессе обучения модели.

```

1 {"Loss/scores/fake": {"num": 8000, "mean": -1.3127416271343828,
2 "std": 1.1875860488528838},
3 "Loss/signs/fake": {"num": 8000, "mean": -0.735,
4 "std": 0.6780671058236051},
5 "Loss/G/loss": {"num": 4000, "mean": 1.6901547474563121,
6 "std": 0.9005358968033085},
7 "Loss/scores/real": {"num": 4248, "mean": 1.394200870111276,
8 "std": 1.3514650773845382},
9 "Loss/signs/real": {"num": 4248, "mean": 0.6822033898305084,
10 "std": 0.7311624545227712},
11 "Loss/D/loss": {"num": 4000, "mean": 0.7285748132839799,
12 "std": 0.43899729894092265},
13 "Loss/r1_penalty": {"num": 248, "mean": 0.04055439657907212,
14 "std": 0.038737182034400026},
15 "Loss/D/reg": {"num": 248, "mean": 0.13382950768760976,
16 "std": 0.12783269644602357}

```

При сравнении со значениями в предыдущем периоде, было выявлено, что В целом, значения потерь показывают, что модель, вероятно, обучается эффективно, с хорошими показателями как для генератора, так и для дискриминатора. Но при этом высокие стандартные отклонения во многих метриках могут указывать на то, что процесс обучения может быть нестабильным или что модель сталкивается с трудностями в определенных итерациях. Поэтому на данный момент обучение модели все еще продолжается.

На данный момент модель генерирует фото подобного типа:

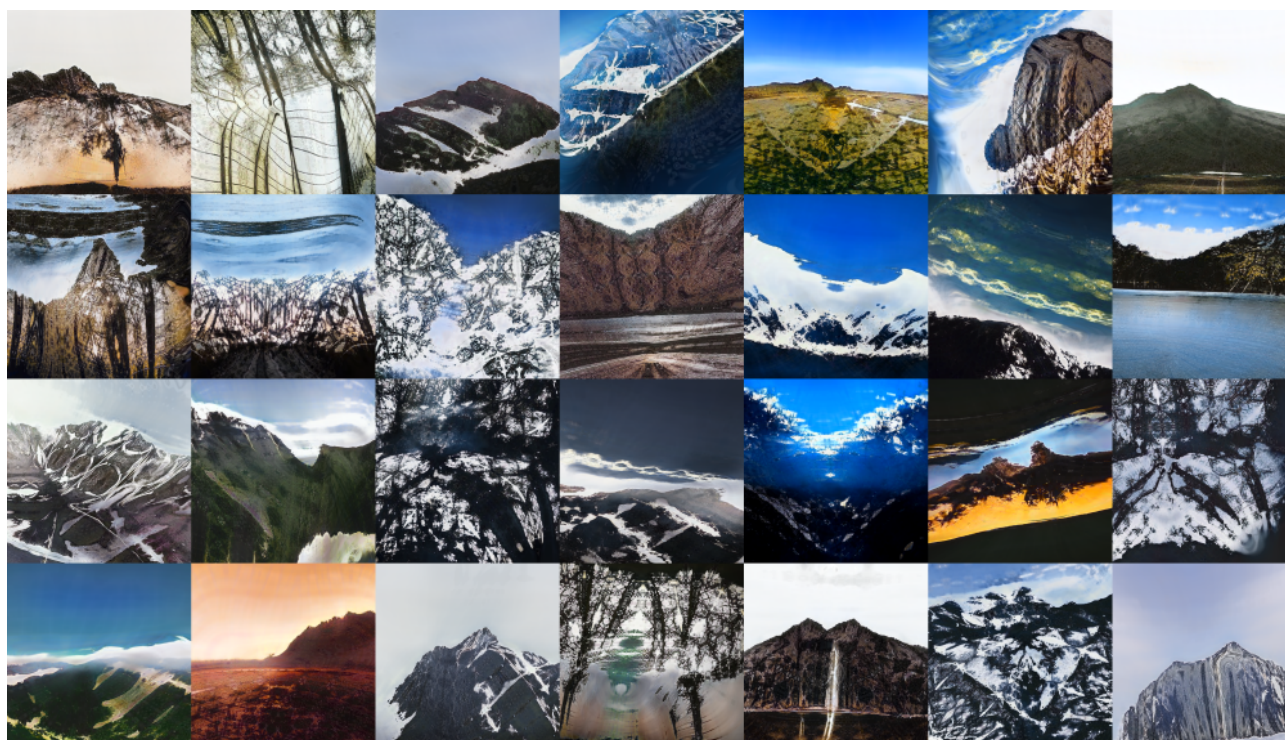


Рисунок 4 — Примеры сгенерированных изображений

Стоит отметить, что изначально была предпринята попытка создать свою GAN, но из-за отсутствия ресурсов и неудовлетворительного результата было принято решение дообучить модель.

**Заключение.** В ходе работы все задачи были выполнены, все цели достигнуты. Было создано веб-приложение, предоставляющее пользователю уход за растением по его запросу. Также в бета-тесте пользователю могут быть доступны дополнительные функции, но поскольку продукт скорее на стадии MVP или же минимально жизнеспособного продукта, они могут быть доступны и сейчас, особенно если рассматривать проект не как коммерческий

продукт. Проект все еще можно улучшить, и этот процесс все еще происходит. Например, модель постепенно обучается, изменяется дизайн, улучшаются функции чата.