

Полученные результаты показывают, что при распознавании изображений дефектов сверточная сеть достаточно эффективна, что достигается благодаря ее устойчивости к искажениям входных данных, а так же делает ее подходящим выбором архитектуры для решения задач обнаружения и классификации дефектов.

В дальнейших исследованиях предполагается исследовать возможности применения адаптивных сети на основе системы нечеткого вывода, для увеличения точности распознавания дефектов, а так же повысить устойчивость системы к разнообразным искажениям входных данных.

#### Литература

1. Bobryakov A., Borisov V., Misnik A. and Prokopenko S. Approaches to the Implementation of Information-Analytical Processes in Complex Technical-Organizational Systems // 2020 V International Conference on Information Technologies in Engineering Education ( Inforino ), 2020, pp. 1-5, doi: 10.1109/Inforino48376.2020.9111833.
2. Николенко С., Кадурич А., Архангельская Е.. Глубокое обучение // Серия «Библиотека программиста». СПб. : Питер, 2018.
3. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. // Серия «Библиотека программиста». СПб.: Питер, 2018. — 576 с.
4. Ян Эрик Содем Программирование компьютерного зрения на языке Python / пер.с англ. Слинкин А.А – М.: ДМК Пресс,2016-312с.

*Е.А. Якимов, к.т.н.; В.Т. Садовский, ст. преп., А.С. Барыгин, А.А. Плотников  
(Белорусско-Российский университет, г. Могилев, Беларусь)*

### **ВЕБ-ПЛАТФОРМА ДЛЯ УДАЛЕННОГО ИЗУЧЕНИЯ УЧЕБНЫХ ДИСЦИПЛИН ПРОФЕССИОНАЛЬНОЙ ПОДГОТОВКИ**

В настоящее время для удаленного обучения студентов выпускающей кафедры «Автоматизированные системы управления» используется платформа Moodle, обеспечивающая удаленный доступ к учебным материалам и реализующая функцию контроля знаний обучаемых. Однако Moodle не поддерживает интеграцию виртуальных лабораторий, что не позволяет обучаемым в достаточной мере получать необходимые им практические знания и навыки.

Основная особенность разработанной веб-платформы – возможность интеграции различных виртуальных лабораторий, которые позволяют выполнять специальные практические задания и, таким образом, закреплять теоретические знания на практике. Такой способ удаленного обучения позволит повысить эффективность процесса удаленного обучения студентов, а также сделать его более разнообразным и интересным [1, 2].

Виртуальная лаборатория предоставляет студентам комплекс задач из различных предметных областей, виртуальные инструменты для формализации условий процесса, средства для решения проблемы; преподавателям – постоянный контроль, диагностику освоения материала. Таким образом, обучаемые самостоятельно могут формировать практические умения и навыки в удобное для них время, не ограничивая себя временем и территориальной отдаленностью от образовательной организации.

Особый интерес представляют веб-лаборатории с удаленным доступом. Они реализуются путем доработки программного обеспечения лаборатории с

одним лабораторным сервером. В этом случае студенты получают доступ к оборудованию с помощью любого персонального компьютера, имеющего выход в интернет. Именно такие виртуальные лаборатории проще всего интегрировать в разрабатываемую веб-платформу, потому что они уже разработаны в виде веб-приложений. В приложении В представлены примеры реализованных студентами кафедры АСУ виртуальных лабораторий.

При разработке веб-платформы были сформулированы следующие задачи:

- организация хранения информации об учебных дисциплинах (различного рода файлы, лабораторные работы, лекции и др.);
- организация быстрого доступа к материалам учебных дисциплин;
- реализация возможности тестирования обучаемых с автоматической оценкой результатов прохождения тестов;
- реализация возможности формирования отчетов по результатам тестирования, что позволяет снизить нагрузку преподавателей.
- поддержка возможности интеграции различных виртуальных лабораторий;
- возможность администрирования и управления всеми объектами, относящимися к веб-приложению (создание, редактирование и удаление различных объектов, таких как тест, лабораторная работа, лекция и др.).

При разработке серверной части веб-платформы использован язык программирования Python, в частности, фреймворк Django Rest Framework. В качестве сервера базы данных используется PostgreSQL. Для разработки графического интерфейса используется язык программирования JavaScript, в частности, фреймворк Vue.js, а также язык разметки Html и каскадные таблицы стилей CSS.

Веб-платформа имеет модульную структуру. Например, модуль работы с учебными дисциплинами (рис. 1), модуль работы с тестами (рис. 2). Основные действия, реализуемые в этих модулях – создание, редактирование и удаление объектов.

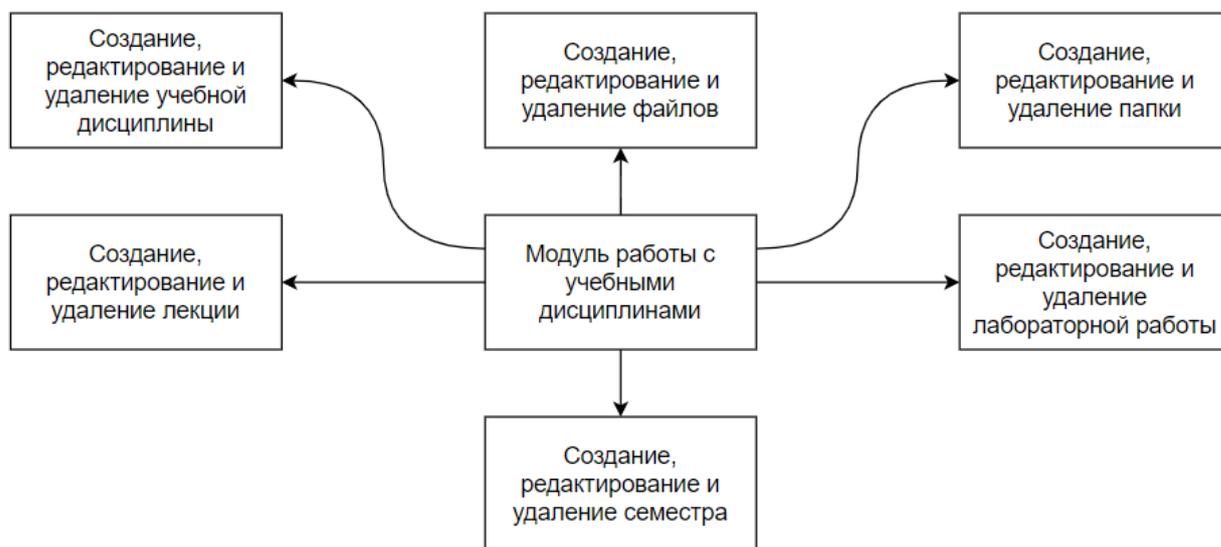


Рисунок 1 – Структура модуля работы с учебными дисциплинами

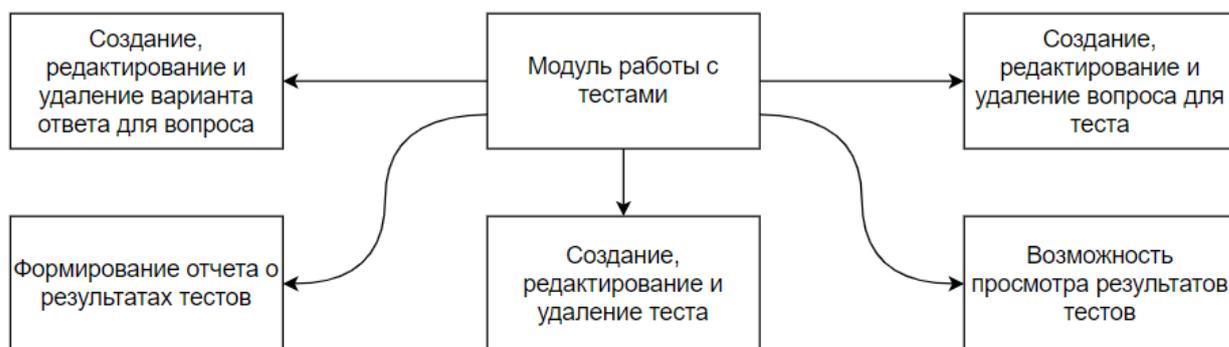


Рисунок 2 – Структура модуля работы с тестами

Кроме разработанных модулей используются внешние библиотеки на языке Python:

- Django – веб-фреймворк, предоставляющий функционал для разработки веб-приложений;
- Django Rest Framework – дополнение к Django, предоставляет функционал для создания restful API приложений;
- Pillow – позволяет работать с изображениями, в приложении используется для добавления изображений в тесты;
- Psycopg2 – предоставляет функционал для работы с PostgreSQL путем написания кода на Python.

При входе в систему с ролью преподавателя пользователю доступны все функции обучаемого и открыт доступ к администрированию. В режиме администрирования можно добавлять, изменять, удалять как устройства, так и команды с подкомандами для устройств виртуальных лабораторий.

Модульная структура приложения обеспечивает возможность интеграции внешних виртуальных лабораторий, позволяет достичь независимости отдельных модулей друг от друга, а также позволяет вести их коллективную разработку.

К разработчикам внешних модулей для серверной части предъявляются следующие требования:

- опыт работы с языком программирования Python;
- опыт работы с библиотекой Django с надстройкой Django Rest Framework;
- опыт работы с реляционными базами данных, в частности, PostgreSQL;
- опыт работы с приложениями, реализованными с помощью клиент-серверной архитектуры;
- опыт работы с системами контроля версий, в частности, GitHub.

Для клиентской части разработчик интегрирования внешних модулей должен обладать опытом работы с языковым средством JavaScript, а также иметь опыт работы с библиотекой Vue.js.

Доступ к внешней виртуальной лаборатории осуществляется с помощью добавления доменного имени разрабатываемой программной системы в реестр имен, имеющих доступ к внешней виртуальной лаборатории. Для обеспечения безопасности необходимо использовать секретный ключ доступа для каждого

пользователя. Секретный ключ позволит идентифицировать пользователя ИС, а также обеспечит синхронизацию между пользователями двух систем.

Модули для серверной части доступны в корневом каталоге платформы `asu_app_api/asu_app/api`. Данная директория содержит все модули разрабатываемой системы. Для добавления нового модуля необходимо создать папку с именем этого модуля. Далее создать файлы, необходимые для реализации функционала модуля. На рисунке 3 представлен пример файлов модуля «Console».

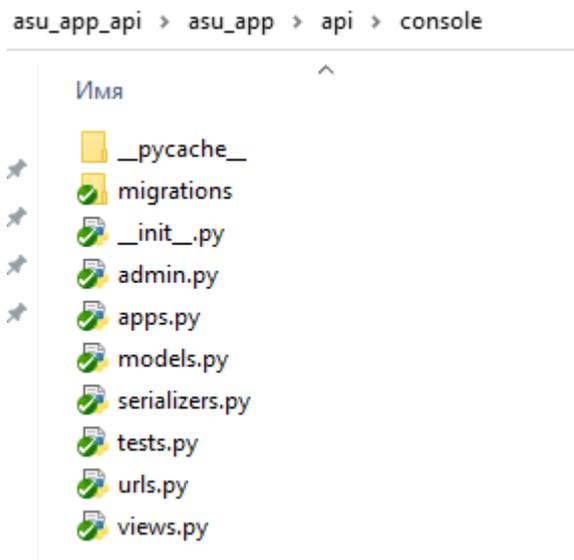


Рисунок 3 – Пример файлов модуля «Console»

Папка «migrations» содержит в себе набор файлов-миграций для базы данных. Файлы-миграции используются для изменения или создания таблиц. Файл «\_\_init\_\_.py» используется для обозначения видимости модуля внутри разрабатываемой системы. Файлы «admin.py» и «apps.py» используются для настройки режима администрирования разрабатываемой системы. Файл «models.py» содержит в себе набор классов, отвечающих за взаимодействие серверной части и базы данных. Файл «serializers.py» содержит в себе код для форматирования данных об объектах системы в JSON формат для отображения их клиентской частью. За передачу данных между серверной частью и клиентской частью отвечает код, содержащийся в папке «view.py». Код маршрутизации модуля содержится в файле «urls.py». Файл «tests.py» используется для проведения автоматизированного тестирования.

После написания необходимого модуля на серверной стороне необходимо добавить модуль в настройках веб-платформы. Для этого в директории основного проекта `asu_app/asu_app` в файле «settings.py» необходимо в переменную «INSTALLED\_APPS» добавить имя модуля. Содержимое переменной «INSTALLED\_APPS»:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',
```

```

'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'rest_framework',
'rest_framework.authtoken',
'corsheaders',
'django_filters',
'nested_admin',
'api.users',
'api.subjects',
'api.groups',
'api.simple_tests',
'api.console',

```

```
]
```

После изменений в файле «settings.py» для доступности функционала нового модуля необходимо добавить код маршрутизации модуля в код маршрутизации всей платформы. Код маршрутизации платформы находится в файле `asu_app/asu_app/urls.py`. В данном файле необходимо изменить переменную «`urlpatterns`». В переменную необходимо добавить путь к маршрутизации модуля. Содержимое переменной «`urlpatterns`»:

```

urlpatterns = [
    path('', include('api.users.urls')),
    path('', include('api.groups.urls')),
    path('', include('api.subjects.urls')),
    path('', include('api.console.urls')),
    path('', include('api.simple_tests.urls')),

```

```
]
```

После всех необходимых настроек серверной части для отображения данных для пользователя необходимо добавить новый модуль для клиентской части. Для этого необходимо создать шаблон страницы, написанный с использованием библиотеки `Vue.js`. После создания его необходимо переместить в директорию клиентской части в папку `src/components`. Данный шаблон страницы должен содержать набор функций, отвечающий за передачу и прием данных с серверной части.

После создания шаблона страницы, его необходимо добавить в систему маршрутизации клиентской части. Для этого необходимо перейти в папку `src/router`, в файле `index.js` добавить новую запись в переменную `routes`.

В случае корректного добавления модулей для клиентской и серверной части при входе на веб-платформу будет доступна возможность работы с новым модулем.

#### Литература

1. Шабров, Н. Н. Программно-аппаратные комплексы виртуального окружения – ключевые компоненты технологий виртуального инжиниринга / Н. Н. Шабров // *CAD/CAM/CAE Observer*. – 2016. – № 3. – С. 83–86.
2. Веб-приложение для обучения администраторов компьютерных сетей предприятий и организаций / Е. А. Якимов [и др.] // *Энергетика, информатика, инновации – 2019* (электроэнергетика, электротехника и теплоэнергетика, математическое моделирование и информационные технологии в производстве, микроэлектроника и оптоэлектроника): сб. трудов IX-ой Междунар. науч.-техн. конф. В 2 т. Т 1; Смоленск, 17–18 октября 2019 г. – Смоленск: филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске, 2019. – С. 346–349.