

ной стороны, обеспечило качество изображений – правильный поворот головы и правильную освещенность. В то же время, многие фотографии были в единственном экземпляре, что оказалось не всегда достаточно для точного извлечения дескрипторов лица. Также недостатком использования нейронных сетей явилось то, что при добавлении нового лица в базу данных приходилось переобучать всю сеть заново.

Для более точного извлечения дескрипторов лица было принято решение использовать 3D-алгоритмы распознавания, использующие информацию о глубине и кривизне поверхности, в отличие от систем двухмерного распознавания, использующих признаки, основанные на яркости пикселей изображения и удаленности разных точек на лице (у каждого лица оно индивидуальное). Такие трехмерные дескрипторы имеют большую точность в описании особенностей поверхности; лучше подходят для описания свойств лица в областях щек, лба и подбородка; и, что очень важно, инвариантны к ракурсу съемки и освещению.

Нельзя не отметить, что главные трудности при распознавании лиц создает проблема освещенности изображения, положение головы, скорость работы алгоритмов, а также, во многих случаях, частичное закрытие лиц прической, соседями по аудитории или некоторыми посторонними объектами (например, медицинской маской).

Т. Е. Козляк, В. И. Шилко
(ГрГУ имени Я. Купалы, Гродно)

Науч. рук. **А. М. Кадан**, канд. техн. наук, доцент

СОЗДАНИЕ ИГРОВЫХ БОТОВ ДЛЯ СОРЕВНОВАНИЙ ПО ПРОГРАММИРОВАНИЮ ЗАДАЧ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Развитие методов искусственного интеллекта (AI) породило новый вид соревнований - чемпионаты по программированию AI. Участники таких соревнований реализуют игровые стратегии (программы-боты), которые состязаются друг с другом в виртуальном мире. Данный тип соревнований предоставляет возможность проверить свои навыки программирования, запрограммировав AI-стратегию, управляющую юнитом в специальном игровом окружении. В каждой игре участвуют стратегии как минимум двух участников. Стратегия, набравшая

большее количество очков, побеждает. Игры также могут закончиться ничьей, если все стратегии набрали одинаковое количество очков.

Среди подобных соревнований [1] достаточно широко известны такие, как AI Cup – открытый чемпионат по программированию искусственного интеллекта компании Mail.Ru Group [2]; Google AI Challenge – международный конкурс программирования искусственного интеллекта [3], начатый Клубом Компьютерных наук университета Ватерлоо. Активно обсуждаются аспекты программирования AI на ряде технических форумов, в частности [4].

Технология проведения соревнований по программированию искусственного интеллекта предполагает проведения турниров в несколько этапов, которым предшествует квалификация в Песочнице. В ходе каждого этапа игровому боту присваивается и изменяется некоторое значение рейтинга, показывающего, насколько успешно его стратегия участвует в играх с соперниками. Песочница используется не только для определения предварительного уровня игры (рейтинга) бота перед отправкой его для участия в следующих этапах соревнований. Также боты отправляются в песочницу для проверки правильности сделанных изменений их игровых стратегий в процессе их разработки и отладки.

Участнику соревнований предоставляется программный пакет, содержащий модули, описывающие виртуальный игровой мир в виде объектно-ориентированной модели (для обеспечения простого расширения и модификации). Также игроку доступно приложение игрового симулятора (для проведения отладки на локальном компьютере) и метод, в котором должна быть реализована логика стратегии его игрового бота. Такому методу должна быть доступна информация о текущем состоянии игры и отладочный интерфейс. Также в состав пакета входит объект, представляющий базовую стратегию (используемую при отладке в роли «противника»).

Время в игре дискретное и поделено на «тики». В начале каждого тика, игровой симулятор передает состояние виртуального мира ботам участников, затем получает от них действия и обновляет состояние мира в соответствии с этими действиями и правилами игры. Затем процесс повторяется для следующего тика с обновленным состоянием. Длительность игры ограничена, но игра может закончиться, если все стратегии игроков «упали»: завершились в результате ошибок времени выполнения или из-за потери взаимодействия с игровым сервером; а также, если стратегия превысила ограничения по времени или памяти.

С целью получения навыков по разработке инструментария для проведения соревнований по программированию искусственного интеллекта, был спроектирован и реализован базовый программный пакет на языке Python. Основные его классы – класс Game, представляющий игру нескольких (неограниченного количества) ботов, и класс Bot – представляющий базовый игровой Бот. Структура классов представлена в таблицах 1 и 2.

Таблица 1 – Структура класса Game

Game	
- bots	список ботов
+ gameArea	параметры виртуального игрового мира
+ end	признак завершения игры
+ winner	ссылка на бот-победитель
+ max_timer	предел времени, отведенного на игру
- current_timer	текущее значение таймера
+ func_game_over()	условие завершения игры
+ game_show()	визуализация состояния игровой среды
+ game_stepShow()	визуализация шага состояния игровой среды
+ game_over()	проверка завершения игры
+ game_check(player)	контроль состояния игры после хода бота player
+ calc_winner()	определение победителя
+ game_end()	вывод победителя
+ game_step()	шаг игры - боты ходят согласно списку bots
+ game_loop()	цикл игры - до завершения

Таблица 2 – Структура класса Bot

Bot	
+ name	имя бота
+ points	очки, набранные ботом в игре
+ visible	признак визуализации действий бота
+ nextStep()	базовая стратегия

Модель игрового мира описывается дополнительными классами и структурами, в зависимости от особенностей игры.

В зависимости от вида игры, требуется создать новый класс игры на основе класса Game и переопределить его методы func_game_over(), определяющий условие завершения игры, и методы game_show(), game_stepShow(), реализующие вывод протокола игры. А также создать интеллектуального бота (интеллектуальную стратегию). Для этого наследуется класс IBot от Bot, в котором переопреде-

ляется метод nextStep(), определяющий следующий шаг бота на основе анализа текущего состояния виртуального игрового мира.

Данный подход использован при разработке виртуальных игровых миров для программирования AI в классических антагонистических играх, как имеющих оптимальные стратегии, так и не имеющих их. Предполагается его использование в соревнованиях по направлению «Интеллектуальная защита» в рамках Олимпиады студентов и школьников по криптографии и защите информации (Junior.Crypt), проводимой кафедрой системного программирования и компьютерной безопасности ГрГУ им.Янки Купалы с 2010 года.

Литература

1. Competitions and prizes in artificial intelligence [Электронный ресурс] / Wikipedia, the free encyclopedia. – URL: https://en.wikipedia.org/wiki/Competitions_and_prizes_in_artificial_intelligence. – Дата доступа: 20.03.2021.

2. AI Cup – соревнование по программированию искусственного интеллекта [Электронный ресурс] / CodeCraft 2020. – URL: <https://russianaicup.ru/>. – Дата доступа: 20.03.2021.

3. Google AI challenge [Электронный ресурс] / Wikipedia, the free encyclopedia. – URL: https://en.wikipedia.org/wiki/AI_Challenge. – Дата доступа: 20.03.2021.

4. Игровая логика и ИИ [Электронный ресурс] / Технический форум. GameDev.ru — Разработка игр. – URL: <https://gamedev.ru/code/forum/?ai>. – Дата доступа: 20.03.2021.

Ю. В. Кузнецов

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **М. А. Подалов**, ст. преподаватель

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УДАЛЕННОГО УПРАВЛЕНИЯ АНТЕННОЙ

В рамках работы была разработана автоматизированная система удаленного управления антенной. Процесс управления системой реализован через мобильное приложение.

Одной из наиболее интересных и активно развивающихся концепций является Интернет вещей (англ. Internet of Things, IoT) – концеп-