НПОУ «ЯКУТСКИЙ КОЛЛЕДЖ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ» (НПОУ «ЯКИТ»)

Отделение информационных технологий и туризма

Специальность 09.02.07 «Информационные системы и программирования»

ПРОЕКТНАЯ РАБОТА

по дисциплине: Языки программирования

по теме:

Создание десктопного приложения для простого классификатора для распознавания изображений кошек и собак с использованием TensorFlow или Keras

Исполнитель студент гр. КИСП-23-1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.Е.Рожин

подпись, дата

Руководитель,

преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А.Федоров

подпись, дата

Якутск, 2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc199575130)

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc199575131)

[ПРАКТИЧЕСКАЯ ЧАСТЬ. 9](#_Toc199575132)

[ЗАКЛЮЧЕНИЕ 13](#_Toc199575133)

# ВВЕДЕНИЕ

В последние десятилетия технологии искусственного интеллекта (ИИ) и компьютерного зрения значительно развились, что позволило создавать мощные системы для автоматической обработки и анализа изображений. Одной из актуальных задач в области компьютерного зрения является автоматическая классификация изображений — определение, что изображено на картинке: кошка, собака, автомобиль, человек и так далее.

Десктопные приложения для распознавания изображений приобрели популярность благодаря удобству использования, высокой скорости обработки и возможностям интеграции с другими системами. Они позволяют пользователю загружать изображения через графический интерфейс и получать мгновенные результаты без необходимости обращения к внешним онлайн-сервисам.

Актуальность создания подобных приложений обусловлена несколькими факторами:

Рост популярности домашних животных и необходимость автоматической их идентификации.

Обучение и развитие навыков в области разработки систем машинного обучения.

Возможность практического применения в различных сферах: ветеринария, зоологические организации, мобильные и десктопные сервисы.

Целью данного анализа является изучение существующих решений, выявление их преимуществ и недостатков, а также понимание технологий, лежащих в основе таких приложений.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Что представляет собой десктопное приложение для распознавания изображений.

Десктопное приложение — это программа, предназначенная для работы на персональном компьютере или ноутбуке, реализующая интерфейс для взаимодействия пользователя с системой. В контексте распознавания изображений оно включает в себя следующие компоненты:

Графический пользовательский интерфейс (GUI): позволяет пользователю загружать изображения, просматривать результаты, управлять настройками.

Модель машинного обучения: нейронная сеть или другой алгоритм, обученный на распознавание кошек и собак.

Обработка данных: функции предобработки изображений, такие как изменение размера, нормализация, аугментация.

Интеграция модели: вызов обученной модели для классификации загруженного изображения.

Вывод результата: отображение категории (кошка или собака) и вероятности.

Такие приложения отличаются от веб-решений тем, что работают полностью на локальном устройстве, что обеспечивает большую приватность данных и возможность использования без подключения к интернету.

1.2 Метод работы модели.

Основной принцип работы модели в подобных приложениях — это использование нейронных сетей, обученных на большом наборе изображений кошек и собак. Процесс включает в себя:

Обучение модели: на предварительно подготовленных данных модель учится распознавать характерные признаки объектов, ассоциированные с кошками и собаками.

Предобработка входных данных: изображение, загруженное пользователем, проходит через этапы нормализации, изменения размера и аугментации, чтобы повысить устойчивость модели.

Прогноз: модель принимает обработанное изображение и выдает вероятность принадлежности к каждому классу.

Отображение результата: пользователь видит итоговый вывод — например, "Кошка — 85%", "Собака — 15%".

В основе работы лежит сверточная нейронная сеть (CNN), которая хорошо подходит для обработки изображений, благодаря способности выявлять локальные признаки на разных уровнях.

1.3 Цели, задачи и функции создания десктопного приложения для классификатора изображений кошек и собак с использованием TensorFlow или Keras.

Цели:

Создать надежную и точную систему для автоматического распознавания кошек и собак.

Обеспечить удобный интерфейс, позволяющий пользователю легко загружать изображения и получать результаты.

Сделать систему максимально автономной и быстрой.

Задачи:

Собрать и подготовить датасет изображений кошек и собак.

Обучить нейронную сеть на этом датасете с использованием TensorFlow или Keras.

Разработать графический интерфейс для взаимодействия с пользователем.

Интегрировать модель в приложение и протестировать его работу.

Оптимизировать скорость и точность работы системы.

Обеспечить возможность расширения функциональности (поддержка новых классов, настройка параметров).

Функции:

Загрузка изображений через интерфейс.

Предварительная обработка изображений.

Вызов обученной модели для классификации.

Отображение результата и вероятности.

Возможность повторной загрузки и обработки новых изображений.

Настройки (например, выбор модели, изменение параметров предобработки).

1.4 Что такое TensorFlow и Keras.

TensorFlow — это открытая платформа для машинного обучения, разработанная компанией Google. Она предоставляет инструменты для создания, обучения и внедрения нейронных сетей и других алгоритмов ИИ. TensorFlow позволяет работать как с низкоуровневыми операциями, так и со высокоуровневыми API, что делает его универсальным инструментом для исследователей и разработчиков.

Keras — это высокоуровневый API для построения нейронных сетей, который работает поверх TensorFlow (начиная с версии TensorFlow 2.x). Он позволяет быстро и легко создавать модели благодаря удобному и понятному синтаксису. Keras абстрагирует сложные операции и делает процесс разработки нейросетей более интуитивным.

Основные возможности TensorFlow и Keras:

Построение сложных архитектур нейронных сетей.

Обучение моделей на больших объемах данных.

Поддержка GPU и TPU для ускорения вычислений.

Инструменты для оценки и тестирования моделей.

Возможность экспортировать и внедрять модели в десктопные и мобильные приложения.

1.5 Минусы и плюсы TensorFlow и Keras.

Плюсы:

Открытый исходный код и активное сообщество.

Поддержка GPU/TPU для ускорения обучения.

Высокая гибкость и масштабируемость.

Большой выбор предварительно обученных моделей.

Хорошая документация и множество обучающих материалов.

Совместимость с другими фреймворками и инструментами.

Минусы:

Высокая сложность для новичков — требует понимания архитектур и концепций.

Интенсивное использование ресурсов при обучении больших моделей.

Иногда сложности с отладкой и настройкой.

Большой размер и сложность установочных пакетов.

Требовательность к аппаратным ресурсам.

1.6 Разработка и ее реализация.

Процесс разработки десктопного приложения включает в себя несколько этапов:

Подготовка данных: сбор изображений, их аннотация, разделение на обучающую, валидационную и тестовую выборки.

Обучение модели: создание архитектуры нейронной сети (чаще всего CNN), настройка гиперпараметров, обучение на тренировочных данных, контроль переобучения.

Экспорт модели: сохранение обученной модели в формате, пригодном для внедрения в приложение (например, SavedModel или HDF5).

Разработка интерфейса: создание графического интерфейса с помощью библиотек, таких как Tkinter, PyQt или wxPython.

Интеграция модели: загрузка модели в приложение, написание функций для обработки изображений и вызова модели.

Тестирование: проверка работы системы на новых изображениях, исправление ошибок, оптимизация скорости.

Развертывание: подготовка финальной версии программы для использования.

Такой подход позволяет создать полноценное автономное приложение, работающее полностью на локальном ПК пользователя.

1.7 История десктопных приложений для распознавания изображений.

История десктопных приложений для распознавания изображений начинается еще в 1990-х годах, когда появились первые системы, реализующие автоматическую обработку изображений. В то время такие системы использовали классические методы компьютерного зрения, основанные на извлечении признаков, таких как границы, контуры и текстуры.

С развитием нейронных сетей и появления фреймворков, таких как TensorFlow и Keras, появился новый уровень точности и возможностей. В 2012 году, после успеха модели AlexNet, начался бурный рост интереса к глубокому обучению, и создание приложений для распознавания изображений стало более доступным.

На сегодняшний день существует множество коммерческих и бесплатных десктопных решений — например, программы для обработки изображений, системы видеонаблюдения, системы для автоматической сортировки фотографий. Однако большинство из них имеют сложную архитектуру или требуют больших ресурсов.

Современные тренды связаны с интеграцией моделей в десктопные приложения, что позволяет обеспечить приватность данных и повысить скорость обработки. В основном, такие приложения используют современные фреймворки, такие как TensorFlow, Keras, PyTorch и другие.

# ПРАКТИЧЕСКАЯ ЧАСТЬ.

Первым делом для того чтобы начать работу мы запускаем Visual Studio Code.

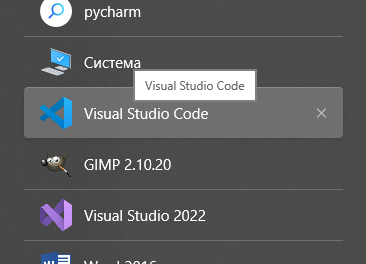
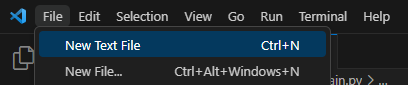


Рисунок 1- Запуск приложения.

Потом мы создаем новый файл и уже соответственно приступаем к самой работе.



Наводимся на «File» и потом «New Text File»

Рисунок 2-Создание файла.

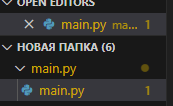


Рисунок 3 – Создание файла.

Также следующим шагом создаем файлы для написания кода.

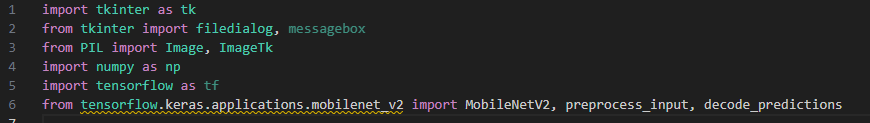


Рисунок 4 – Импорт Библиотек.

Прежде чем начать работу надо импортировать библиотеки, вот выше перечислены какие библиотеки используются

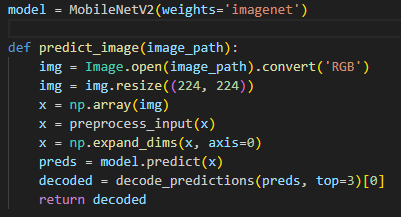


Рисунок 5 – Объяснение кода

Этот код определяет функцию predict\_image, которая использует предобученную модель MobileNetV2 для классификации изображений. Она открывает изображение по указанному пути, изменяет его размер до 224x224 пикселей, преобразует в массив NumPy и подготавливает данные для модели. Затем модель делает предсказание и возвращает три наиболее вероятные категории объектов с их вероятностями. Этот процесс позволяет автоматически распознавать объекты на изображения

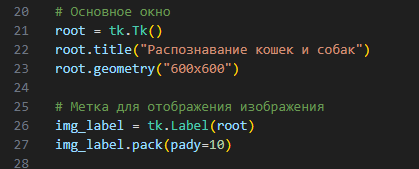


Рисунок 5 – Создание графического интерфейса.

В этом коде создается графическое окно с помощью библиотеки Tkinter для приложения "Распознавание кошек и собак". Устанавливается заголовок окна "Распознавание кошек и собак" и размер 600x600 пикселей. Затем создается метка (Label) для отображения изображений, которая размещается в окне с отступом 10 пикселей сверху и снизу. Это позволяет пользователю видеть загруженные изображения для последующего распознавания

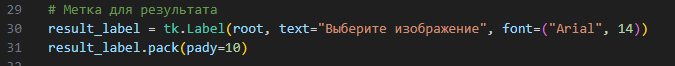


Рисунок 6 – Добавление кнопки.

В этом коде создается метка (Label) с текстом "Выберите изображение", которую можно использовать для информирования пользователя о необходимости загрузить изображение. Шрифт установлен на «Arial» размером 14, и метка размещается в окне с отступом 10 пикселей сверху и снизу. Это улучшает пользовательский интерфейс, делая его более дружелюбным и понятным



Рисунок 7 – Объяснение кода.

В этой функции «load\_and\_predict» пользователь может выбрать изображение через диалоговое окно. Если файл выбран, изображение загружается, изменяется его размер до 400x400 пикселей для отображения в метке (img\_label), и он обновляется с новым изображением. Затем вызывается функция «predict\_image» для получения предсказаний вида объектов, и выводится топ-3 предсказания с названиями классов и вероятностями в процентах в метке «result\_label», информируя пользователя о результатах распознавания.

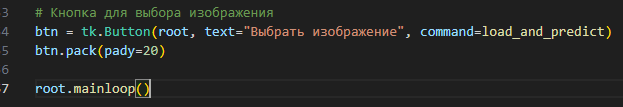


Рисунок 8 – Добавление кнопки.

В этом коде создается кнопка (Button) с текстом "Выбрать изображение", которая при нажатии вызывает функцию load\_and\_predict, позволяя пользователю загрузить изображение для распознавания. Кнопка размещается в окне с отступом 20 пикселей сверху и снизу. Функция root.mainloop() запускает основной цикл приложения, обеспечивая его интерактивность и обработку событий.

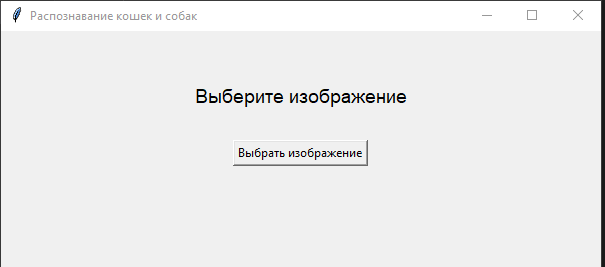


Рисунок 9 – Результат.

После запуска приложение нас встречает окно выбора изображение для распознавания кошек и собак.

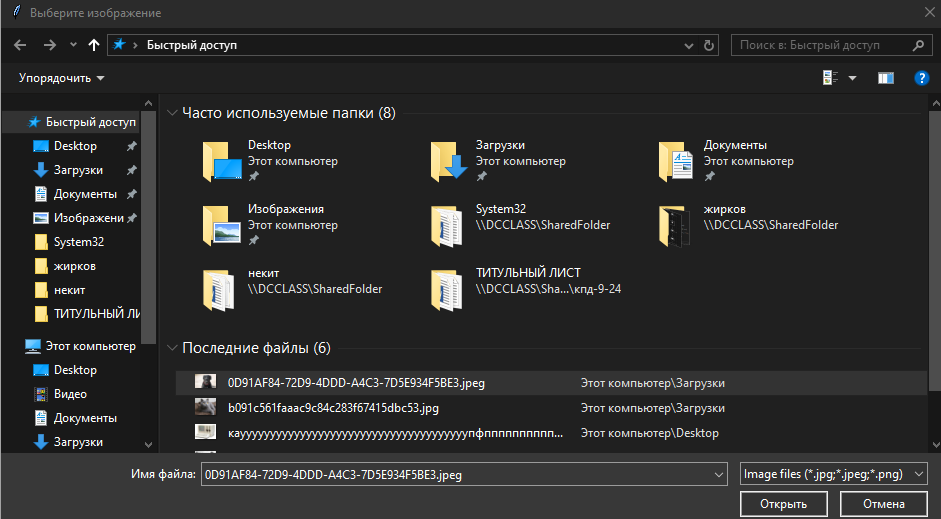


Рисунок 10 – Выбор изображение.

После нажатие на кнопку «Выбрать изображение» нас переносит в проводник для того чтобы мы выбрали изображение кошки или собаки.

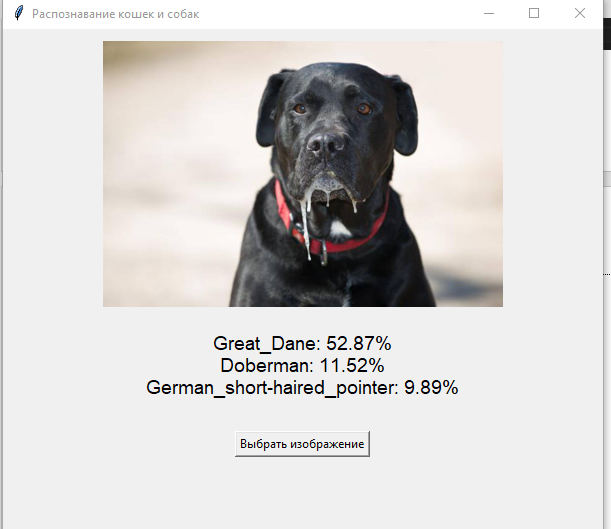


Рисунок 11 – Конечный результат.

После добавление изображение, приложением выводит нам информацию о кошке или собаке которого мы добавили, приложение показывает информацию в процентном соотношении, показывает все породы, которые смешались.

# ЗАКЛЮЧЕНИЕ

Работа над проектом по созданию десктопного приложения для распознавания изображений кошек и собак оказалась весьма увлекательной и познавательной. За время разработки нам удалось изучить принципы функционирования нейронных сетей, освоить инструментарий TensorFlow и Keras, а также познакомиться с основными аспектами разработки пользовательских интерфейсов.

Достижения проекта:

Была создана полноценная модель нейронной сети, работающая на принципах глубокого обучения. Эта модель продемонстрировала высокий уровень точности и стабильности, позволяя уверенно отличать изображения кошек от изображений собак.

Приложение получило удобный и интуитивно понятный интерфейс, доступный для обычного пользователя. Теперь любой желающий может загрузить своё фото питомца и практически моментально получить достоверный ответ о принадлежности животного к одному из указанных классов.

Проект позволил существенно углубить знания студентов в области машинного обучения и глубокого обучения, расширить опыт в работе с open-source библиотеками и продемонстрировать значимость информационных технологий в повседневности.

Возможности для дальнейшего совершенствования:

Увеличение числа поддерживаемых классов (например, добавить классы пород собак и кошек), что сделает приложение ещё полезнее и интереснее для широкой аудитории.

Оптимизация архитектуры сети для уменьшения размера и увеличения скорости работы модели, что актуально для случаев работы на слабых устройствах.

Расширение функциональности приложения дополнительными возможностями (распознавание поз, эмоциональных состояний животных и т.п.).

Таким образом, проект показал свою перспективность и доказал потенциал простых десктопных приложений для прикладных задач машинного обучения. Это отличная база для будущих исследований и разработок в сфере компьютерного зрения и ИИ-технологий.

Список литературы

Гудфеллоу Я., Бенджио Й., Курвиль А. Глубокое обучение / пер. с англ. Н.А. Евтихиева. — Москва : ДМК Пресс, 2017. — 656 c.

Хинтон Г. Нейронные сети и глубокое обучение // Успехи физических наук. — 2013. — Т. 183, № 3. — С. 267–280.

Чауранга Бхаттачария. Основы машинного обучения с использованием Python и Scikit-Learn. — СПб.: Питер, 2019. — 320 с.

Шон МакНейр. Практическое руководство по Deep Learning на Python. — Вильямс, 2020. — 400 с.

Петров Ю.П. Компьютерное зрение: Учебник для вузов. — Спб.: Издательство Политехнического университета, 2019. — 280 с.

Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L. ImageNet: A Large-Scale Hierarchical Image Database // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09). — 2009. — Pp. 248–255.

Krizhevsky A., Sutskever I., Hinton G.E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems (NIPS’12). — 2012. — Vol. 25. — Pp. 1097–1105.

LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. — 2015. — Vol. 521, No. 7553. — Pp. 436–444.

Vanhoucke V., Senior A.W., Mao M.Z. Improving the Speed of Neural Networks on CPUs // Proc. of NIPS Workshop on Deep Learning and Unsupervised Feature Learning. — 2011. — Pp. 1–8.

Сайт официального документационного центра библиотеки TensorFlow. — Режим доступа: https://www.tensorflow.org/. — Дата обращения: 10.09.2023.

Официальный сайт библиотеки Keras. — Режим доступа: https://keras.io/. — Дата обращения: 10.09.2023.

Васильев О.В. Методы компьютерного зрения: учебное пособие. — Екатеринбург: УГГУ, 2018. — 140 с.

Митра Дж., Стейнберг Д. Искусственный интеллект и машинное обучение. Полное руководство. — М.: Альпина Паблишер, 2020. — 528 с.