

ИСПОЛЬЗОВАНИЕ PYTHON И GOOGLE COLAB ПРИ РАЗРАБОТКЕ ПРОТОТИПОВ СИСТЕМ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Аннотация. В статье рассматриваются возможности языка Python и облачной среды Google Colab для быстрой разработки прототипов систем распознавания изображений. Показаны преимущества открытых библиотек OpenCV, PyTorch и torchvision для обработки изображений, организации данных и построения моделей глубокого обучения. Особое внимание уделяется подходу переноса обучения при решении задач классификации.

Ключевые слова: Python, Google Colab, компьютерное зрение, обработка изображений, перенос обучения, нейронные сети, прототипирование, классификация.

Разработка систем распознавания изображений требует удобной и гибкой среды, которая позволяет быстро проходить весь цикл – от загрузки данных до обучения модели и анализа результатов [1]. Одним из наиболее популярных и эффективных решений сегодня является сочетание языка программирования Python и облачной платформы Google Colab. Благодаря бесплатному доступу к GPU, удобному интерфейсу Jupyter Notebook и богатой экосистеме библиотек это сочетание идеально подходит для создания демонстрационных и учебно-исследовательских прототипов систем компьютерного зрения.

Главным преимуществом Python является наличие мощных открытых библиотек. Библиотека OpenCV предоставляет широкие возможности по фильтрации, сегментации, изменению контраста и предварительной обработке изображений. Для задач глубокого обучения чаще всего используются PyTorch вместе с torchvision, которые содержат готовые архитектуры нейронных сетей, средства аугментации данных и инструменты для переноса обучения [2]. Эти инструменты позволяют исследователю сосредоточиться на логике задачи, а не на низкоуровневой реализации.

Особенно удобна в Google Colab подготовка и организация набора данных. С помощью библиотек torchvision.datasets и ImageFolder изображения легко структурируются по папкам-классам, автоматически загружаются и разделяются на обучающую и тестовую выборки. Такой подход очень практичен при разработке систем классификации промышленных объектов (например, канистр с разными этикетками), когда данные естественно организованы по категориям [3].

Важным достоинством связки Python + Colab является отличная визуализация. Библиотеки Matplotlib и Seaborn позволяют наглядно отображать процесс обработки изображений, примеры правильной и ошибочной классификации, графики обучения и матрицы ошибок. Это особенно ценно при подготовке материалов для выпускной квалификационной работы и научных публикаций – всё можно делать в одном ноутбуке и сразу делиться ссылкой [4].

Python и Google Colab отлично подходят для быстрого прототипирования. Разработчик может за несколько часов перейти от идеи к работающей модели, используя готовые шаблоны и обученные сети (ResNet, EfficientNet, Vision Transformer). Возможность бесплатного использования GPU существенно ускоряет эксперименты, что критично при ограниченном времени на дипломный проект [5].

При создании систем классификации изображений широко применяется перенос обучения (transfer learning). Предобученная на большом датасете (ImageNet) модель



адаптируется под небольшое количество новых классов. Это позволяет добиться высокой точности даже при ограниченном объёме собственных данных и значительно сократить время обучения [6].

Хотя Python-решения чаще используются на этапе исследования и прототипирования, чем для прямого промышленного развёртывания, именно на стадии проверки гипотез и создания демонстрационных моделей они показывают максимальную эффективность. Полученный прототип легко переносится в другие среды при необходимости.

Таким образом, сочетание Python и Google Colab представляет собой современную, гибкую и функционально богатую платформу для разработки прототипов систем распознавания изображений. Наличие мощных библиотек, удобной визуализации, бесплатного доступа к вычислительным ресурсам и поддержки переноса обучения делает этот инструментарий особенно ценным в образовательной и исследовательской практике. В контексте создания интеллектуальных систем контроля продукции он обеспечивает скорость, наглядность и высокое качество получаемых результатов.

Список литературы:

1. Шакирьянов Э. Д. Компьютерное зрение на Python. Первые шаги. М.: Лаборатория знаний, 2021.
2. Рашка С., Мирджалили В. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2. М.: ДМК Пресс, 2023.
3. VanderPlas J. Python Data Science Handbook. 2nd ed. Sebastopol: O'Reilly Media, 2023.
4. Chollet F. Deep Learning with Python. 2nd ed. Shelter Island: Manning Publications, 2021.
5. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge: MIT Press, 2016.
6. Форсайт Д. А., Понс Ж. Компьютерное зрение. Современный подход. М.: Вильямс, 2004.

