АО «АЛМАТИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет инженерных и информационных технологий

Кафедра компьютерной инженерии

ПРОЕКТИРОВАНИЕ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

ЛЕКЦИЯ № 9

AI-системы для NLP, Computer Vision и мультимодальных данных

Теоретико-практический конспект

Составитель: Алтыбай А.

Кафедра компьютерной инженерии

Алматы, 2026

Аннотация и назначение лекции

Лекция 9 раскрывает тему «AI-системы для NLP, Computer Vision и мультимодальных данных» в контексте разработки промышленной AI-системы. Материал опирается на тематический план дисциплины и соединяет определения, принципы проектирования, математические оценки, проектные артефакты, практический код и аргументированный анализ рисков. Вопросы построены вокруг решений, которые предстоит принимать инженеру, а не только вокруг названий алгоритмов. [1, 2]

План занятия

1. Модальности и контракты
2. NLP preprocessing
3. Текстовые эмбеддинги
4. CV preprocessing
5. Детекция и классификация
6. Аудио и временные сигналы
7. Мультимодальная архитектура
8. Векторные индексы
9. Практические сценарии, контроль качества и проектные задания.

Ожидаемые результаты обучения

По завершении студент способен:

1. объяснить и аргументированно применить принцип «Модальности и контракты» к инженерному решению;
2. объяснить и аргументированно применить принцип «NLP preprocessing» к инженерному решению;

3. объяснить и аргументированно применить принцип «Текстовые эмбеддинги» к инженерному решению;
4. объяснить и аргументированно применить принцип «CV preprocessing» к инженерному решению;
5. объяснить и аргументированно применить принцип «Детекция и классификация» к инженерному решению;
6. объяснить и аргументированно применить принцип «Аудио и временные сигналы» к инженерному решению;
7. реализовать упрощённый проверяемый прототип и сформулировать критерии его приемки.
8. указать угрозы достоверности результатов и предложить проверки до эксплуатации.

Сквозной кейс. В течение лекции рассматриваем сервис обработки обращений и фотографий. Существенные условия: отсутствие текста, неизвестные форматы изображения. Основная проверяемая характеристика: качество на модальных срезах. Числа в расчётных примерах учебные и не являются результатами реальных измерений.

1. Базовые теоретические положения из конспекта курса

Ниже представлены исходные положения учебно-методического конспекта, связанные с данной темой. Следующие разделы раскрывают их с инженерными деталями и проверяемыми примерами. [1, 2] Цель лекции. показать особенности систем для текста, изображений и мультимодальных данных, включая preprocessing, model serving, latency и выбор специализированных архитектур.

Ключевые понятия. NLP, tokenization, CV, augmentation, OCR, multimodal model, encoder, inference pipeline, batching, preprocessing parity.

1.1 NLP-система

В текстовой системе pipeline обычно включает очистку/нормализацию, токенизацию, модель и постобработку. Для трансформеров tokenization является частью контракта модели: несовместимая версия tokenizer может разрушить качество даже при правильных весах.

Системные требования зависят от длины последовательности, batch size и модели. Длинный контекст резко увеличивает стоимость вычислений и память.

1.2 Computer Vision система

CV pipeline включает декодирование изображения, изменение размера, нормализацию, иногда детекцию/segmentation, модель и postprocessing. В production важны вариации камер, освещения, разрешения, компрессии и ориентации.

Аугментации применяются при training и не должны слепо переноситься в inference. При edge deployment важны размер модели, quantization и аппаратная совместимость.

1.3 Мультимодальные системы

Мультимодальные модели объединяют текст, изображения, аудио и другие источники. Архитектура должна синхронизировать модальности и определить поведение при отсутствии одной из них. Например, карточка товара может иметь изображение и текст, но часть объектов содержит только одно поле.

При проектировании следует учитывать различную частоту обновления и стоимость обработки каждой модальности.

1.4 Preprocessing parity и контракты

Для специализированных моделей особенно важно зафиксировать точную версию preprocessing: tokenizer, vocabulary, image resize/crop, normalization constants, audio sample rate. Эти элементы нужно версионировать вместе с моделью.

Полезно публиковать модель как пакет с manifest: входная схема, preprocessing, версия весов, выходная схема, допустимые диапазоны и примеры.

1.5 Производительность inference

Для тяжелых моделей применяют dynamic batching, quantization, compilation, caching, distillation и специализированные runtimes. Оптимизация должна измеряться end-to-end: ускорение ядра модели бесполезно, если основное время занимает декодирование или сетевой обмен.

В реальных системах p95/p99 latency часто важнее среднего значения.

1.6 Выводы

- NLP/CV pipeline включает preprocessing и postprocessing как часть модели продукта.
- Мультимодальная система должна определять поведение при неполных модальностях.
- Tokenizer и preprocessing необходимо версионировать вместе с весами.
- End-to-end latency важнее локальной скорости одного слоя.

1.7 Контрольные вопросы

- Почему tokenizer является частью контракта NLP-модели?
- Какие виды shift характерны для CV?

- Что нужно учитывать при отсутствии одной из модальностей?
- Что такое dynamic batching?
- Почему следует измерять p95/p99 latency?

2. Систематическое изложение темы и инженерные решения

2.1 Модальности и контракты

Текст, изображение, аудио и видео имеют разные структуры, размерности и источники шума. Модальный контракт задает кодировку, длину, разрешение и допустимые форматы. [2]

Проверяемый пример. Для входа JPG или текстового сообщения определите два валидатора и унифицированную схему ответа.

Вопрос для обсуждения. Как проверить выполнение такого требования: Для входа JPG или текстового сообщения определите два валидатора и унифицированную схему ответа.

2.2 NLP preprocessing

Unicode normalization, токенизация, учет языка и усечение текста влияют на семантику. Агрессивная очистка может удалить отрицания и важные символы. [3]

Проверяемый пример. Сравните прогноз до и после удаления слова «не» из пользовательского отзыва.

Вопрос для обсуждения. Как проверить выполнение такого требования: Сравните прогноз до и после удаления слова «не» из пользовательского отзыва.

2.3 Текстовые эмбеддинги

Плотные семантические векторы облегчают поиск и классификацию. Их нужно оценивать на предметной лексике и языках пользователей. [4]

Проверяемый пример. Сравните поиск похожих русских и казахских обращений.

Вопрос для обсуждения. Как проверить выполнение такого требования: Сравните поиск похожих русских и казахских обращений.

2.4 CV preprocessing

Декодирование, resize, нормализация и сохранение aspect ratio влияют на качество. Аугментации применяют только в соответствующей train-процедуре. [5]

Проверяемый пример. Проверьте, искажает ли squashing объект на изображениях разных пропорций.

Вопрос для обсуждения. Как проверить выполнение такого требования: Проверьте, искажает ли squashing объект на изображениях разных пропорций.

2.5 Детекция и классификация

Классификация относится к изображению целиком, детекция локализует объекты, сегментация прогнозирует маску. Выбор задачи определяет разметку и метрики. [6]

Проверяемый пример. Не используйте image-level accuracy для задачи поиска дефектов на конвейере.

Вопрос для обсуждения. Как проверить выполнение такого требования: Не используйте image-level accuracy для задачи поиска дефектов на конвейере.

2.6 Аудио и временные сигналы

Частота дискретизации, windowing и спектрограммы определяют, какая информация доступна модели. Случайная нарезка соседних окон может породить leakage. [1]

Проверяемый пример. Разбейте аудиозаписи по говорящим, а не случайным коротким фрагментам.

Вопрос для обсуждения. Как проверить выполнение такого требования: Разбейте аудиозаписи по говорящим, а не случайным коротким фрагментам.

2.7 Мультимодальная архитектура

Раннее объединение признаков и позднее слияние предсказаний отличаются устойчивостью к пропуску модальности. Cross-attention изучает зависимости между ними. [2]

Проверяемый пример. Для пары «товарное фото + описание» укажите fallback при отсутствии изображения.

Вопрос для обсуждения. Как проверить выполнение такого требования: Для пары «товарное фото + описание» укажите fallback при отсутствии изображения.

2.8 Векторные индексы

Косинусная близость, L2 и approximate nearest neighbors дают разные свойства поиска. Индекс обновляют согласованно с encoder version. [3]

Проверяемый пример. Покажите, почему замена embedding-модели требует перестроения индекса.

Вопрос для обсуждения. Как проверить выполнение такого требования:

Покажите, почему замена embedding-модели требует перестроения индекса.

2.9 Batching и динамические размеры

Для текстов используются padding и attention mask, для изображений — унификация размерности. Пакетирование влияет на throughput и хвост latency. [4]

> **Проверяемый пример.** Примените dynamic batching с ограничением максимального ожидания запроса.

Вопрос для обсуждения. Как проверить выполнение такого требования: Примените dynamic batching с ограничением максимального ожидания запроса.

2.10 Оценка модальных срезов

Средняя метрика скрывает влияние языка, разрешения, освещения, длительности записи. Необходима проверка на срезах с достаточной выборкой. [5]

> **Проверяемый пример.** Создайте таблицу F1 отдельно по двум языкам и трем диапазонам длины текста.

Вопрос для обсуждения. Как проверить выполнение такого требования: Создайте таблицу F1 отдельно по двум языкам и трем диапазонам длины текста.

2.11 Приватность контента

Изображения и аудио часто содержат лица, голоса и геометки; хранение промежуточных embeddings тоже требует оценки рисков. [6]

Проверяемый пример. Укажите сроки хранения загруженных файлов и правила обезличивания логов.

Вопрос для обсуждения. Как проверить выполнение такого требования: Укажите сроки хранения загруженных файлов и правила обезличивания логов.

2.12 Интерфейс с пользователем

UX должен сообщать доверительный уровень, ограничение модели и возможность исправления результата, не обещая абсолютной достоверности. [1]

Проверяемый пример. Разработайте экран ошибки обработки невалидного аудио и сценарий ручной проверки.

Вопрос для обсуждения. Как проверить выполнение такого требования: Разработайте экран ошибки обработки невалидного аудио и сценарий ручной проверки.

3. Математические инструменты и количественные критерии

Для перехода от неформального решения к проверяемой постановке введём следующие величины.

$$H_{\text{out}} = \left\lfloor \frac{H + 2P - D(K-1) - 1}{S} \right\rfloor + 1 \tag{1}$$

$$\text{TopKRecall} = \frac{\#\{\text{релевантные объекты в топ-}K\}}{\#\{\text{все релевантные объекты}\}} \tag{2}$$

Первая формула задаёт высоту карты признаков свёртки, вторая — полноту поиска; метрики отличаются для разных модальностей и этапов.

При выборе числового критерия нужно отличать обучающую функцию потерь, продуктовую ценность и эксплуатационные ограничения. Первую можно оптимизировать дифференцированием, вторая определяется процессом принятия решения, а третьи проверяются нагрузочными испытаниями, политиками безопасности и мониторингом.

3.1 Пошаговый расчёт на учебных данных

Рассмотрим две конфигурации одной системы. Вариант A имеет условную оценку качества $q_A = 0.86$ и задержку $t_A = 90$ мс; вариант B — $q_B = 0.89$ и $t_B = 280$ мс. Предположим, что требование заказчика задаёт $q \geq 0.85$ и $p95 \leq 150$ мс. Тогда A удовлетворяет обоим условиям, тогда как B нарушает второе. Следовательно, выбор B только потому, что его offline-метрика выше, не является обоснованным. Числа иллюстративны; реальный вывод требует измерений при одинаковой нагрузке и одинаковом наборе данных.

$$\text{accept}(A) = \mathbf{1}\{0.86 \geq 0.85 \wedge 90 \leq 150\} = 1,$$
$$\text{accept}(B) = \mathbf{1}\{0.89 \geq 0.85 \wedge 280 \leq 150\} = 0.$$

3.2 Как интерпретировать число, а не только вычислить его

Сначала следует проверить единицы измерения, состав выборки и смысл агрегирования. Например, p95 на спокойном стенде нельзя переносить на рабочую нагрузку с иным числом одновременных запросов. Затем исследуют чувствительность к изменениям входных данных и неопределённость оценок. Наконец, фиксируют последствия порога: какую ошибку он предотвращает, а какую может увеличить. Этот подход делает количественный критерий частью инженерной аргументации, а не украшением отчёта. [1, 2]

4. Практическое программирование на Python

Пример предназначен для демонстрации ключевого инженерного принципа. Он не претендует на обработку реального production-трафика. Запускать код следует в изолированном окружении и фиксировать версии библиотек, если они используются. Вывод зависит от указанных входных данных; для стохастических алгоритмов он может меняться.

4.1 Основной воспроизводимый пример

```python
import torch
from torch import nn

# Text example: padded integer token IDs, batch-first tensor.
encoder = nn.Embedding(num_embeddings=5000, embedding_dim=32,
                       padding_idx=0)
classifier = nn.Sequential(nn.Linear(32, 16), nn.ReLU(), nn.Linear(16, 2))
tokens = torch.tensor([[4, 5, 0], [7, 8, 9]])
mask = (tokens != 0).unsqueeze(-1)
emb = encoder(tokens)
pooled = (emb * mask).sum(dim=1) / mask.sum(dim=1).clamp(min=1)
logits = classifier(pooled)
print("logits shape:", logits.shape)
```

Проверка результатов. Сначала убедитесь, что код завершился без исключений; проверьте формы массивов, схему результата и отсутствие недопустимых значений. Если используются внешние данные, включите в отчёт способ их получения и идентификатор версии. Измените один входной параметр и объясните, почему наблюдаемый результат изменился или остался прежним.

4.2 Компактная проверка контракта или метрики

```python
# Check the modality contract at the boundary.
import torch
x = torch.randn(4, 3, 224, 224)
```

```
assert x.ndim == 4 and x.shape[1] == 3
print("batch shape:", tuple(x.shape))
```

При самостоятельном повторении предложите минимум один негативный тест: некорректное значение, пустой вход, несовместимую версию схемы либо нарушение SLO. Переход от успешного демонстрационного запуска к надежной системе требует именно таких проверок.

5. Разбор трёх инженерных ситуаций

5.1 Ситуация 1: Модальности и контракты

Заказчик использует сервис обработки обращений и фотографий. Новое требование заставляет рассмотреть принцип «Модальности и контракты». Основная неопределённость состоит в том, как одновременно обеспечить качество на модальных срезах, не допустить неизвестные форматы изображения и сохранить возможность изменения модели.

Постановка. Текст, изображение, аудио и видео имеют разные структуры, размерности и источники шума. Модальный контракт задает кодировку, длину, разрешение и допустимые форматы.

Действие инженера. Для входа JPG или текстового сообщения определите два валидатора и унифицированную схему ответа.

Альтернативы. Сравните три подхода: (A) простое правило без ML, (B) модульное ML-решение с отдельной проверкой входных данных и (C) интегрированный сервис со сложной моделью. Для каждого укажите потребность в данных, ожидаемые инженерные риски, возможность тестирования и минимальную инфраструктуру. Таблица ниже задаёт форму отчёта, а не готовый рейтинг.

Вариант	Данные и зависимости	Проверка	Основной риск
Правила	Фиксированные условия	Набор тест-кейсов	Ограниченная гибкость
ML-модуль	Версионированные признаки	Offline и API tests	Drift и skew
Сложный сервис	Несколько компонентов	End-to-end и нагрузка	Каскадный отказ

Проверка решения. Подготовьте две позитивные и две негативные пробы; задайте метрику успеха, базовый сценарий и критерий остановки внедрения. Обсудите, какие допущения останутся непроверенными даже после удачного теста.

5.2 Ситуация 2: Аудио и временные сигналы

Заказчик использует сервис обработки обращений и фотографий. Новое требование заставляет рассмотреть принцип «Аудио и временные сигналы». Основная неопределённость состоит в том, как одновременно обеспечить качество на модальных срезах, не допустить неизвестные форматы изображения и сохранить возможность изменения модели.

Постановка. Частота дискретизации, windowing и спектрограммы определяют, какая информация доступна модели. Случайная нарезка соседних окон может породить leakage.

Действие инженера. Разбейте аудиозаписи по говорящим, а не случайным коротким фрагментам.

Альтернативы. Сравните три подхода: (A) простое правило без ML, (B) модульное ML-решение с отдельной проверкой входных данных и (C) интегрированный сервис со сложной моделью. Для каждого укажите потребность в данных, ожидаемые инженерные риски, возможность тестирования и минимальную инфраструктуру. Таблица ниже задаёт форму отчёта, а не готовый рейтинг.

Вариант	Данные и зависимости	Проверка	Основной риск
Правила	Фиксированные условия	Набор тест-кейсов	Ограниченная гибкость
ML-модуль	Версионированные признаки	Offline и API tests	Drift и skew
Сложный сервис	Несколько компонентов	End-to-end и нагрузка	Каскадный отказ

Проверка решения. Подготовьте две позитивные и две негативные пробы; задайте метрику успеха, базовый сценарий и критерий остановки внедрения. Обсудите, какие допущения останутся непроверенными даже после удачного теста.

5.3 Ситуация 3: Приватность контента

Заказчик использует сервис обработки обращений и фотографий. Новое требование заставляет рассмотреть принцип «Приватность контента». Основная неопределённость состоит в том, как одновременно обеспечить качество на модальных срезах, не допустить неизвестные форматы изображения и сохранить возможность изменения модели.

Постановка. Изображения и аудио часто содержат лица, голоса и геометки; хранение промежуточных embeddings тоже требует оценки рисков.

Действие инженера. Укажите сроки хранения загруженных файлов и правила обезличивания логов.

Альтернативы. Сравните три подхода: (A) простое правило без ML, (B) модульное ML-решение с отдельной проверкой входных данных и (C) интегрированный сервис со сложной моделью. Для каждого укажите потребность в данных, ожидаемые инженерные риски, возможность тестирования и минимальную инфраструктуру. Таблица ниже задаёт форму отчёта, а не готовый рейтинг.

Вариант	Данные и зависимости	Проверка	Основной риск
Правила	Фиксированные условия	Набор тест-кейсов	Ограниченная гибкость
ML-модуль	Версионированные признаки	Offline и API tests	Drift и skew
Сложный сервис	Несколько компонентов	End-to-end и нагрузка	Каскадный отказ

Проверка решения. Подготовьте две позитивные и две негативные пробы; задайте метрику успеха, базовый сценарий и критерий остановки внедрения. Обсудите, какие допущения останутся непроверенными даже после удачного теста.

6. Самостоятельная работа: индивидуальные проектные задания

Для каждого задания студент создаёт отдельный небольшой артефакт: схему, набор тестов, код либо таблицу сравнения, прикладывает наблюдаемый результат и объясняет ограничения.

1. **Модальности и контракты.** Для входа JPG или текстового сообщения определите два валидатора и унифицированную схему ответа. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

2. **NLP preprocessing.** Сравните прогноз до и после удаления слова «не» из пользовательского отзыва. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

3. **Текстовые эмбеддинги.** Сравните поиск похожих русских и казахских обращений. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

4. **CV preprocessing.** Проверьте, искажает ли squashing объект на изображениях разных пропорций. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

5. **Детекция и классификация.** Не используйте image-level accuracy для задачи поиска дефектов на конвейере. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

6. **Аудио и временные сигналы.** Разбейте аудиозаписи по говорящим, а не случайным коротким фрагментам. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

7. **Мультимодальная архитектура.** Для пары «товарное фото + описание» укажите fallback при отсутствии изображения. Сформулируйте

проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

8. **Векторные индексы.** Покажите, почему замена embedding-модели требует перестроения индекса. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

9. **Batching и динамические размеры.** Примените dynamic batching с ограничением максимального ожидания запроса. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

10. **Оценка модальных срезов.** Создайте таблицу F1 отдельно по двум языкам и трем диапазонам длины текста. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

11. **Приватность контента.** Укажите сроки хранения загруженных файлов и правила обезличивания логов. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

12. **Интерфейс с пользователем.** Разработайте экран ошибки обработки невалидного аудио и сценарий ручной проверки. Сформулируйте проверяемое требование, продемонстрируйте альтернативу и приложите протокол хотя бы одного эксперимента.

7. Вопросы для текущего и итогового контроля

1. В чём назначение и ограничения принципа «Модальности и контракты»?

2. Как проверить на практике утверждение: «Для входа JPG или текстового сообщения определите два валидатора и унифицированную схему ответа.»?

3. В чём назначение и ограничения принципа «NLP preprocessing»?

4. Как проверить на практике утверждение: «Сравните прогноз до и после удаления слова «не» из пользовательского отзыва.»?
5. В чём назначение и ограничения принципа «Текстовые эмбеддинги»?
6. Как проверить на практике утверждение: «Сравните поиск похожих русских и казахских обращений.»?
7. В чём назначение и ограничения принципа «CV preprocessing»?
8. Как проверить на практике утверждение: «Проверьте, искажает ли squashing объект на изображениях разных пропорций.»?
9. В чём назначение и ограничения принципа «Детекция и классификация»?
10. Как проверить на практике утверждение: «Не используйте image-level accuracy для задачи поиска дефектов на конвейере.»?
11. В чём назначение и ограничения принципа «Аудио и временные сигналы»?
12. Как проверить на практике утверждение: «Разбейте аудиозаписи по говорящим, а не случайным коротким фрагментам.»?
13. В чём назначение и ограничения принципа «Мультимодальная архитектура»?
14. Как проверить на практике утверждение: «Для пары «товарное фото + описание» укажите fallback при отсутствии изображения.»?
15. В чём назначение и ограничения принципа «Векторные индексы»?
16. Как проверить на практике утверждение: «Покажите, почему замена embedding-модели требует перестроения индекса.»?
17. В чём назначение и ограничения принципа «Batching и динамические размеры»?
18. Как проверить на практике утверждение: «Примените dynamic batching с ограничением максимального ожидания запроса.»?
19. В чём назначение и ограничения принципа «Оценка модальных срезов»?

20. Как проверить на практике утверждение: «Создайте таблицу F1 отдельно по двум языкам и трем диапазонам длины текста.»?
21. В чём назначение и ограничения принципа «Приватность контента»?
22. Как проверить на практике утверждение: «Укажите сроки хранения загруженных файлов и правила обезличивания логов.»?
23. В чём назначение и ограничения принципа «Интерфейс с пользователем»?
24. Как проверить на практике утверждение: «Разработайте экран ошибки обработки невалидного аудио и сценарий ручной проверки.»?
25. Почему высокое offline-качество не гарантирует надежность промышленной AI-системы?
26. Какие артефакты нужны для воспроизводимости и анализа инцидента?

8. Глоссарий ключевых понятий

Модальности и контракты
Текст, изображение, аудио и видео имеют разные структуры, размерности и источники шума.

NLP preprocessing
Unicode normalization, токенизация, учет языка и усечение текста влияют на семантику.

Текстовые эмбеддинги
Плотные семантические векторы облегчают поиск и классификацию.

CV preprocessing
Декодирование, resize, нормализация и сохранение aspect ratio влияют на качество.

Детекция и классификация
Классификация относится к изображению целиком, детекция локализует объекты, сегментация прогнозирует маску.

Аудио и временные сигналы
Частота дискретизации, windowing и спектрограммы определяют, какая информация доступна модели.

Мультимодальная архитектура
Раннее объединение признаков и позднее слияние предсказаний отличаются устойчивостью к пропуску модальности.

Векторные индексы
Косинусная близость, L2 и approximate nearest neighbors дают разные свойства поиска.

Batching и динамические размеры
Для текстов используются padding и attention mask, для изображений — унификация размерности.

Оценка модальных срезов
Средняя метрика скрывает влияние языка, разрешения, освещения, длительности записи.

Приватность контента

Изображения и аудио часто содержат лица, голоса и геометки; хранение промежуточных embeddings тоже требует оценки рисков.

Интерфейс с пользователем

UX должен сообщать доверительный уровень, ограничение модели и возможность исправления результата, не обещая абсолютной достоверности.

9. Итоги и связь со следующей темой

Представленные методы имеют смысл как части целостного управляемого жизненного цикла. Ключевой результат инженерного проектирования состоит не в максимальном числе компонентов или сложности модели, а в убедительной связи требований, проектных решений, измерений и механизмов сопровождения. Дальнейшее изучение курса развивает эти связи на следующем этапе жизненного цикла.

Список литературы

[1] Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. https://www.deeplearningbook.org/.

[2] Vaswani A. et al. Attention Is All You Need. NeurIPS, 2017. https://arxiv.org/abs/1706.03762.

[3] PyTorch Foundation. PyTorch documentation. https://pytorch.org/docs/stable/.

[4] Huyen C. Designing Machine Learning Systems. O'Reilly Media, 2022. https://www.oreilly.com/library/view/designing-machine-learning/9781098107956/.

[5] Murphy K. P. Probabilistic Machine Learning: An Introduction. MIT Press, 2022. https://probml.github.io/pml-book/.

[6] Mitchell M. et al. Model Cards for Model Reporting. FAT*, 2019. https://arxiv.org/abs/1810.03993.