ML детекция спуфинг-атак в области "Liveness Detection”

0. Вводная информация

0.1. Текст задания

Тестовое Задание: Разработка новой архитектуры для детекции спуфинг-атак в области "Liveness Detection"

Общие требования

Мы приглашаем вас создать новую, инновационную архитектуру для решения проблемы детекции спуфинг-атак в рамках задачи "Liveness Detection". Этот проект предполагает разработку нового подхода или модификацию существующих подходов с целью создания уникальной и эффективной системы.

Новизна: Важным условием является то, что предложенный подход или архитектура должны быть новыми и не быть скопированными с открытых ресурсов, таких как GitHub. Комбинация различных подходов допустима, но итоговое решение должно обладать характеристиками научной новизны.

Срок: Ваша задача - предложить концепцию новой архитектуры в течение одной недели.

Цель: Основная цель задания - не обязательно получить решение, превосходящее все существующие, но показать свой инновационный подход к решению задачи, продемонстрировать способность мыслить за рамками привычных подходов и творчески применять свои знания в области компьютерного зрения.

Специфические требования

Архитектура: Предложите концептуальную архитектуру модели, которая могла бы эффективно обрабатывать задачу "Liveness Detection". Ваша архитектура должна

включать в себя: входные данные, слои обработки, функцию потерь и метрики для оценки.

Метрики: Нам интересно услышать ваши предложения по новым или модифицированным метрикам, которые могут помочь в оценке эффективности вашего подхода.

Спуфинг-атаки: Ваша архитектура должна быть специально разработана для обнаружения определенного типа спуфинг-атаки. Выберите тип атаки, который, по вашему мнению, наиболее актуален, и объясните, как ваш подход может справиться с ним.

Оформление результатов: Все результаты должны быть оформлены в репозитории на GitHub. В репозитории должны быть ясно описаны использованные подходы, метрики, принципы работы архитектуры и ее потенциальная эффективность.

0.2. Вводная информация

Атака подделки (spoofing attack) в системах обнаружения живости (liveness detection) является важным компонентом в биометрических системах аутентификации, особенно в случаях, когда для проверки подлинности пользователя используется распознавание лица. Цель обнаружения живости заключается в том, чтобы убедиться, что биометрический образец, который был получен, принадлежит живому человеку, а не искусственной подделке или подставе.

Некоторые типы атак:

1. **Print Attack**: Используется статическое изображение или распечатка лица пользователя в высоком разрешении. Это могут быть печатные изображения или отображаться на экране.
2. **Replay Attack**: Здесь злоумышленники используют предварительно записанные видеоролики или последовательности движений лица, чтобы выдать себя за пользователя.
3. **3D Mask Attack**: 3D-маски или скульптуры лица пользователя для создания реалистичного представления, которое может обойти простые системы основе 2D-изображений..

4. **Other**: Высококачественные маски и макияж, пластические операции, носимые устройства, взлом канала связи или системы хранения эталонных образов.

Некоторые способы защиты:

- **Texture Analysis**: Анализ текстуры лица для обнаружения аномалий, которые могут указывать на фальшивое изображение, таких как неестественные узоры или однородность.
- **Motion Analysis**: Изучение движения в видео, чтобы отличить естественные движения лица от повторяющихся или искусственных, наблюдаемых при повторной атаке.
- **Depth Information**: Использование трехмерной информации о глубине, такой как использование камеры глубины или структурированного света, для определения реальных структур лица и отличия их от двухмерных масок или изображений.
- **Contextual Analysis**: Включение дополнительного контекста, такого как шаблоны поведения пользователя или окружающей среды.
- **Challenge-Response Tests**: Предлагая пользователю случайные вызовы, такие как улыбка, моргание или поворот головы, чтобы обеспечить взаимодействие в реальном времени, а не просто статичное изображение или видео.

1. Цели и предпосылки

1.1. Зачем идем в разработку продукта? Постановка задачи

Системы аутентификации по видео становятся всё более востребованными и требования к их надежности возрастают. Нужно защититься от существующих и потенциальных атак, уменьшив финансовые и репутационные издержки.

В соотвествии с заданием необходимо рассмотреть определенный тип спуфинг-атаки. Рассмотрим отдельное направление - Face Liveness. Этот источник подразделяет атаки на пять типов:

- принт или двухмерное изображение

- маска или трехмерный объект
- специально сделанная высококачественная трехмерная маска
- подделка зашифрованной 3D карты лица, которая используется для идентификации
- перехват и подделка видеопотока, например ззаменой лица (deep fake)

Первые три существуют давно и системы уже должны неплохо с ними справляться. Хотя и наиболее больше количество (по предварительному анализу) приходится всё ещё на них.

Четвертый тип специфичен для определенных систем идентификации.

Сфокусируемся на предотвращении пятого типа, так как он может становиться более популярным при развитии удаленной аутентификации например через веб-камеру или камеру телефона. Существует возможность подменить видеопоток, а с помощью deep fake обойти систему тестов вопрос-ответ. Поэтому защита от такого типа атак актуальна.

Заметим, что рассматриваемые подходы могут быть так же частично или полностью применимы для других типов атак при подготовке соответствующих обучающих данных и своими особенностями (например, определение фрейма фото или пикселей экрана)

Предполагается, что классические методы без машинного обучения не могут справиться с такими атаками, поэтому рассматривается архитектура на ML.

С точки зрения бизнеса успехом будет являться снижение пропусков атак при сохранении доли корректной идентификации, как следствие увеличение финансовой и репутационной выгоды. Стоимость внедрения системы должны быть меньше ожидаемой выгоды.

1.2. Бизнес-требования и ограничения

Интерпретируемость - позволит понять особенности атак и помочь предотвратить их в будущем

Система является критичной - поэтому важна надежность: пилот, ввод в эксплуатацию со страховкой в виде других систем.

1.3. Что входит в скоуп проекта/итерации, что не входит

В рамках итерации (поставленной задачи) входит сбор данных, предобработка, структура модели (слои обработки), функцию потерь и метрики

Не входит в итерацию (задачу) точная постановка бизнес задачи, развертывание и деплой, дообучение, анализ рисков

1.4. Предпосылки решения

Данные - видео

Горизонт прогноза - реальное время

Гранулярность модели - видеопоток небольшой длительности

Типы атак - мы остановились на детекции deep fake

Работа с ошибками - анализ ошибочных ответов с использованием интерпретируемости модели

1.5. Бизнес-метрики

Цена пропуска атаки это прямая угроза, которая может быть выражена в величине штрафа или стоимости закрытой информации

Цена ложной блокировки ниже, может быть выражена как средняя стоимость клиента, умноженная на вероятность потери клиента при ложной блокировке.

$$P = \frac{1}{q}(P_{NF}\widetilde{TP} - P_{FN}FN - \beta P_C FP)$$

P - бизнес метрика, прибыль от внедрения системы в пересчете на одного пользователя, максимизируем

q - количество пользователей

P_FN - цена штрафа за пропуск атаки

\widetilde{TP} - количество верно выявленных атак, которые не были замечены другими системами. Позволяет понять, сколько система добавила ценности к уже существующей.

FN - количество пропусков атак

\beta - вероятность ухода клиента при ложной блокировке

P_c - цена, эквивалентная стоимости потери клиента

FP - количество ложных блокировок

В формуле для простоты не учитывается как повлияет на вероятность ухода пользователя из продукта его неоднократная блокировка. Можем представить, что \beta это некоторая функция от пользователя

2. Методология

2.1. Постановка задачи

Задача классификации (по постановке задачи мы работаем с определенным типом атак и имеем размеченные данные для обучения, поэтому не является задачей выявления аномалий)

2.2 Выбор метрик

Оффлайн-метрики

Предпосылки

- Метрика машинного обучения должно соответствовать бизнес метрике.
- Мы можем сгенерировать достаточное количество примеров атак, поэтому дисбаланса классов в обучающей выборе нету. Будем готовить примерно одинаковое количество примеров для каждого класса.
- В реальном использовании есть большой дисбаланс классов

Преобразуем метрику Accurary и добавим взвешивание ошибок:

$$A = \frac{TP + TN}{TP + TN + FP + FN} = 1 - \frac{FN + FP}{TP + TN + FP + FN}$$

$$A_W = 1 - \frac{\alpha FN + (1-\alpha)FP}{TP + TN + FP + FN},\ \alpha = \frac{P_{FN}}{P_{FN} + \gamma\beta P_C},\ \gamma = 2$$

\gamma введена для компенсации дисбаланса в проде, доля нормальных случаев в проде почти в два раза больше чем в обучении.

Хоть эта метрика и хорошо отражает бизнес метрику, но у неё есть ряд проблем:

- \alpha будет близок к единице и появятся проблемы, специфичные для дисбаланса классов.
- Мы не можем адекватно оценить \alpha
- Сложно интерпретировать метрику.

Раз появляется искусственный дисбаланс, то попробуем повторить операции с F1 (учтем, что \alpha близок к единице):

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FN + FP)}$$

$$F_{1W} = \frac{TP}{TP + \frac{1}{2}(\alpha FN + (1-\alpha)FP)} \approx \frac{TP}{TP + \frac{1}{2}FN} \approx \frac{TP}{TP + FN} \approx Recall$$

Получается, что Recall близок к требуемой метрике. Действительно, при большой цене FN, важно выявить все случаи. Но Recall никак не контролирует FP

[Итоговая метрика]

Поэтому приходим к **Recall@Specificity** > 99.9%: это метрика измеряет способность системы корректно идентифицировать атаку (высокий recall), минимизируя при этом ложноположительные результаты (высокий Specificity).

Specificity нечувствителен к переходу от сбалансированного обучения к несбалансированному проду, контролирует именно процент ложноположительных результатов, хорошо интерпретируется. Кривая Recall - Specificity является монотонной и выбор порога можно производить бинарным поиском, что немного ускорит работу.

Порог Specificity можно выбрать, основываясь на историческом значении метрики.

Функция потерь - логистическая функция.

Дополнительных асимметрий и взвешиваний добавлять не будем. Выбор порога для выбранной метрики решает вопрос разной цены ошибки.

Онлайн-метрики

- количество жалоб о ложной блокировке
- количество выявленных атак по жалобам, анализу действий пользователей и так далее

Технические метрики

Не рассматриваем в итерации

2.3 Определение объекта и таргета

Объект - видео небольшой (определенной) длительности

Таргет - признак, является ли видео с deepfake

2.4 Сбор данных

Собираем видеообразцы и генерируем deep fake. Соблюдаем примерную равную пропорцию классов.

Для исключения смещения нужно использовать как можно больше разных сетей-генераторов.

Формат (качество камеры, света и т.д.) должны соответствовать ожидаемым в работе. Часть видео может быть записана или взята из записей существующей системы, но под неё тоже должны быть сгенерированы атаки.

2.5 Подготовка данных

Используем стандартные методы аугментации

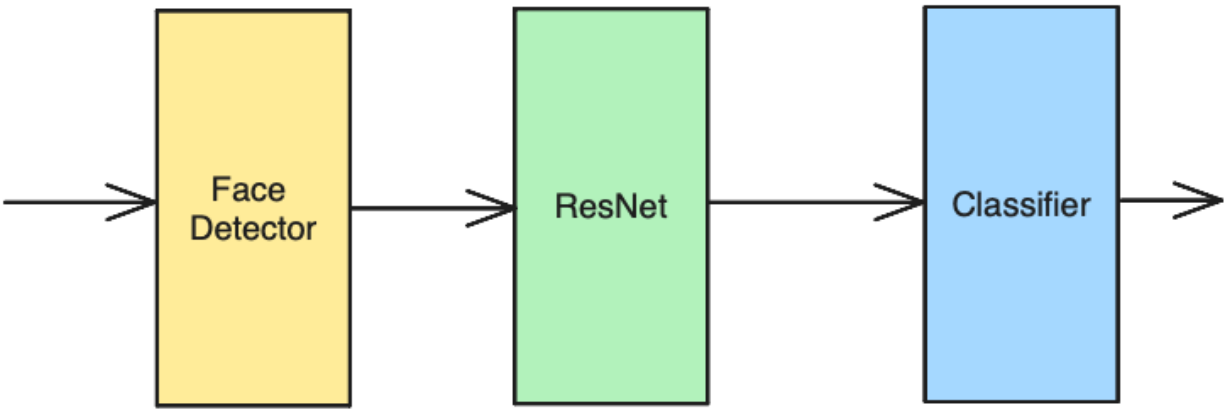
Проверка качества данных (длительность, формат, расширение, содержимое, шумность…)

2.6 Схема валидации

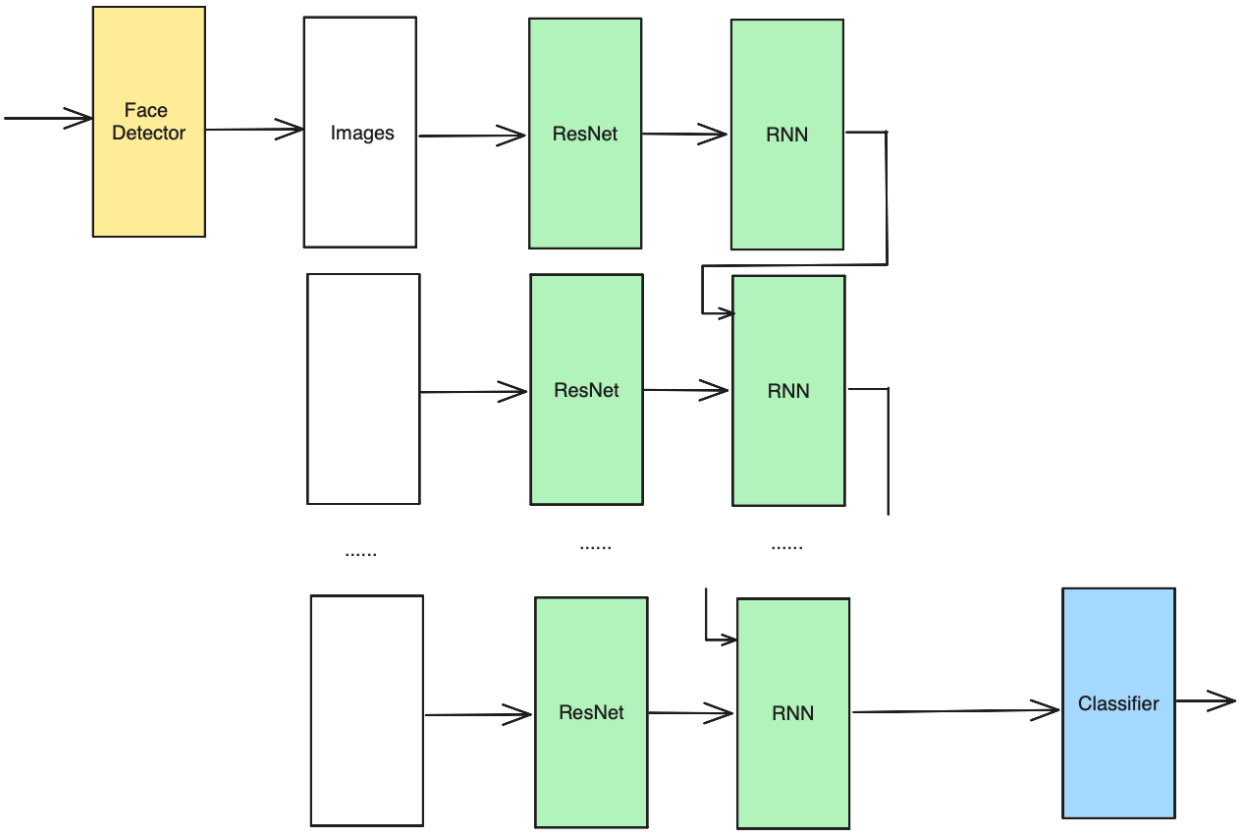
Не рассматриваем в итерации

2.7 Архитектура. От baseline к полноценной модели

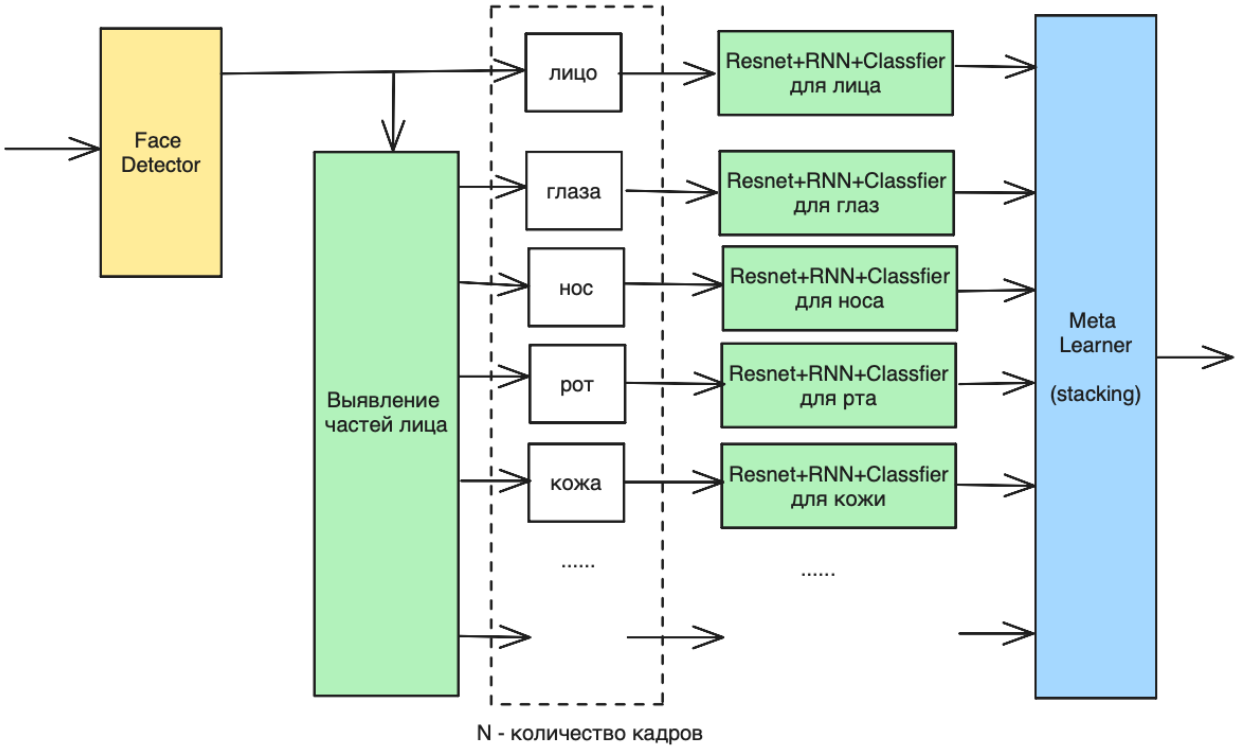
Сделаем простой baseline и будем далее усложнять итерационно



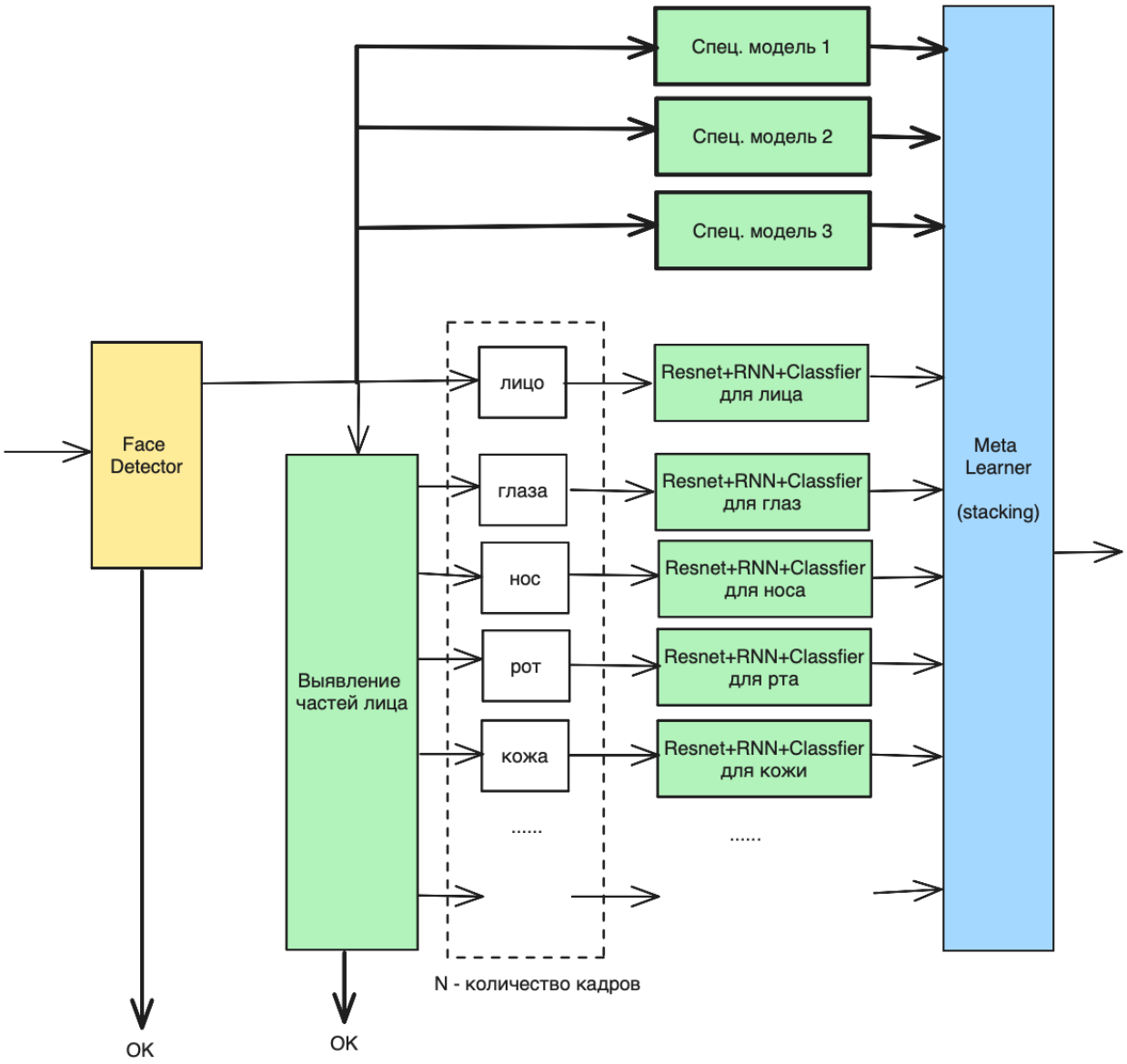
Baseline. Выявляем лицо (найти готовое решение) - предобученная ResNet - классификатор с предпоследнего слоя. Результат может быть как среднее по всем кадрам или один кадр с хорошей детекцией лица



Для обработки последовательности кадров добавим RNN. В дальнейшем будем использовать этот подход для обработки видео. Он также может быть легко адаптирован для потокового видео.



Сделаем выявление частей лица (желательно готовый алгоритм, например на маркерах или уже обученная сеть для сегментации). Отдельные модели будут сфокусированы на конкретной области лица. Результат моделей объединяем через стекинг



Дальнейшее улучшение

Добавлены “Спец. модель”. Каждая из таких моделей настроена на какую-то особенность, например изменение цвета кожи от изменения кровотока периодично с частотой сердечных ударов. Модели так же могут быть настроены на узкоспециализированные обнаруженные типы атак. Специальные модели относятся к будущим улучшениям и их архитектура в этой итерации не прорабатывалась

Дополнительное внимание качеству данных. Блоки выявления лица и частей лица имеют выходы “OK”, которые сигнализируют, что в кадре есть лицо, оно

нормально распознается, достаточного качества, части лица видны и т.д. При несоблюдении дальнейшая проверка нецелесообразна и аутентификация не возможна.

2.8 Анализ ошибок

Следует уделить внимание ошибкам моделей

Про интерпретируемость:

- Отдельные модели выдают свой результат независимо, он может быть проанализирован
- Слои после сверток могут быть интерпретированы, чтобы выявить области с максимальным влиянием
- Для рекуррентных слоев воспользоваться соответствующими методами

3. Подготовка пилота и внедрения

Не рассматриваем в итерации