ИССЛЕДОВАНИЕ УСТОЙЧИВОСТИ ДЕТЕКТОРОВ ОБЪЕКТОВ СЕМЕЙСТВА YOLO К СОСТЯЗАТЕЛЬНЫМ АТАКАМ

DOI: 10.36724/2072-8735-2025-19-6-4-14

Фомичева Светлана Григорьевна,
Санкт-Петербургский государственный университет аэрокосмического приборостроения, г. Санкт-Петербург, Россия,
levikha@mail.ru

Manuscript received 02 May 2025;
Accepted 28 May 2025

Ключевые слова: *детектор объектов, функция потерь, состязательная атака, бюджет атаки, производительность YOLO-детекторов*

В реальных условиях детекторы объектов сталкиваются с различными помехами и шумами, а также вредоносными модификациями, которые могут существенно влиять на их производительность. Такие изменения способны привести к выведению из штатных режимов работы скомпрометированных детекторов, что актуализирует задачи мониторинга существующих и вновь создаваемых YOLO-моделей на устойчивость к состязательным атакам. Целью исследования является оценка устойчивости интеллектуальных детекторов объектов к комплексным состязательным атакам в ходе трансферного обучения. Использовались методы машинного обучения и состязательной защиты. Результаты оценки производительности различных моделей семейства YOLO к состязательным атакам, направленным на градиенты, показали, что наиболее эффективными являются атаки на градиенты терма уверенности функции потерь, приводящие к деградации производительности моделей от 20% при малом бюджете атаки до 100% при больших бюджетах глубоких атак на градиенты. Предложенная в работе тактика комплексной атаки на градиенты модели обосновывают целесообразность применения механизмов состязательного обучения, как обязательной меры защиты, а также дополнительных мер, таких как проведение обучения и эксплуатации верифицированных YOLO-моделей в доверенных среды выполнения. Гарантированно противодействовать глубоким состязательным атакам на градиенты каким-либо одним методом защиты не удается, необходимый и достаточный набор средств защиты определяется спецификой эксплуатации модели.

Информация об авторе:
Фомичева Светлана Григорьевна, *к.т.н., профессор, профессор Санкт-Петербургского государственного университета аэрокосмического приборостроения, г. Санкт-Петербург, Россия*

Для цитирования:
Фомичева С.Г. Исследование устойчивости детекторов объектов семейства YOLO к состязательным атакам // T-Comm: Телекоммуникации и транспорт. 2025. Том 19. №6. С. 4-14.
For citation:
S. G. Fomicheva, "Resistance researching of YOLO object detectors to adversarial attacks," *T-Comm*, 2025, vol. 19, no. 6, pp. 4-14. (*in Russian*)

Введение

К пилотным проектам, реализующим стратегические направления в области цифровой трансформации транспортной отрасли Российской Федерации до 2030 г. относятся проекты по автономному судовождению, беспилотным авиационным системам и беспилотным логистическим коридорам, в функционале которых реализуются задачи компьютерного зрения, и, в частности, задача интеллектуального обнаружения и распознавания объектов. К числу наиболее успешных и широко используемых механизмов решения задач детектирования объектов относится семейство нейросетевых моделей YOLO (You Only Look Once) [1].

Эти модели демонстрируют высочайшую скорость работы, что позволило их массово внедрять в различные высокопроизводительные пограничные (edge) устройства, работающие в режиме реального времени, к числу которых относятся видеокамеры, системы отслеживания траектории движения и другие встраиваемые системы [2-5]. Однако в реальных условиях детекторы объектов сталкиваются с различными помехами и шумами, а также вредоносными модификациями, которые могут существенно влиять на их производительность. Такие изменения способны привести к выведению из штатных режимов работы скомпрометированных детекторов [6-8], что актуализирует задачи оценки существующих и вновь создаваемых YOLO-моделей на устойчивость к состязательным атакам.

Поиск новых атак и лучших стратегий защиты нейросетевых моделей является активной областью исследований. Наличие уязвимостей у YOLO-детекторов зафиксировано в международной базе (например, CVE с номером CVE-2021-31681 [9] − Десериализация уязвимости ненадежных данных в YOLOv3 позволяет злоумышленникам выполнять произвольный код с помощью созданного файла yaml). Уровень критичности данной уязвимости высок и равен 7,8 из 10. Эволюция моделей YOLO вплоть по текущих версий YOLOv11/v13 в основном была направлена на добавление компонент, которые повышают точность обнаружения за счет многомасштабного обучения, фундаментальная архитектура не изменилась и унаследовала те же уязвимости. В частности, YOLO-детектор, по-прежнему, остается подверженным состязательным атакам [7, 8]. Чувствительность архитектурных компонент YOLO-детекторов к случайным и преднамеренным модификациям на данный момент мало изучена, что мотивирует проведение подобных исследований.

В данной работе приведены результаты исследования устойчивости интеллектуальных детекторов объектов семейства YOLO к основным типам состязательных атак: отравление датасетов, на которых проводится обучение YOLO, а также отравление самой модели машинного обучения (ML-модели) YOLO при шумовой и вредоносной модификации ее нейросетевых параметров. Исследовались получившие широкое практическое применение модели YOLOv5, YOLOv8, а также одна из новых моделей YOLOv11 [1].

В отличие от ранее проведенных исследований [10-12], которые в основном направлены на оценку защищенности либо одной конкретной YOLO-модели, либо моделей, решающих предиктивную задачу только классификации, в данной работе представлен ряд результатов для решения задач обнаружения объектов семейством YOLO-моделей и включающих в себя как задачи регрессии (локализации объекта), так и задачи классификации. Для этого предложен и реализован новый подход к формированию комплексных атак, направленных на параметры ограничивающих искомый объект рамок (b-box) и классификационные метки. Эксплуатация атак по данным шаблонам оценена с помощью метрик производительности YOLO-детекторов (mAP, Precision, Recall, F1-Score). Изучение устойчивости YOLO-детекторов позволила выявить их слабые места, и, как следствие, предложить методы повышения их устойчивости к шумам и преднамеренным несанкционированным модификациям.

В данной работе в разделе 1 приводятся в формализованном виде используемые методы и принципы состязательности с акцентом на архитектурные особенности YOLO-моделей. Далее в разделе 2 представлены результаты трансферного обучения YOLO-детекторов объектов на чистых (не состязательных) образцах, в разделе 3 описывается стратегия предложенной комплексной атаки на YOLO-модели. В разделе 4 приводятся результаты оценки производительности различных исмплементаций модели YOLOv11 (Nano, Small, Medium, Large и Extra Large) после их дообучения с состязательными примерами, добавленными в обучающую выборку и последующим переобучением, а также приводятся рекомендации по защите от градиентных состязательных атак.

1 Формальное представление используемых методов и принципов состязательности

Состязательное машинное обучение [13] – это метод машинного обучения, который при обучении использует тактику обмана ML-модели, предоставляя на вход модели как реальные данные (с корректным распределением исходных данных в домене), так и данные, вводящие ML-модель в заблуждение (с искаженным распределением данных в домене). Следовательно, оно включает в себя как генерацию, так и обнаружение состязательных примеров, которые являются входными данными, специально созданными для обмана классификаторов, детекторов и иных типов ML-систем. Следуя подобной тактике ML-модель обучается не только верно классифицировать входные данные и прогнозировать выход, но и правильно обобщать данные. Поэтому состязательное обучение является сложившейся практикой, обязательной в техниках MLSecOps [14]. Для реализации состязательной тактики атак с целью оценки устойчивости ML-модели к верному принятию решений (прогнозу), используют, как указано выше, так называемые состязательные примеры [13].

С формальной точки зрения для некоторой ML-модели f и ее входного сигнала x с истинной меткой y (при решении задачи классификации) состязательным примером является возмущенный входной сигнал x' такой, что $\|x–x'\|$ невелико для некоторой функции расстояния $\|\cdot\|$, и, во-вторых, он классифицирован неправильно так, что для нецелевой атаки $f(x') \neq y$, а для целевой атаки $f(x') = t$ для некоторого выбранного класса $t \neq y$.

При рассмотрении оценки качества моделей требуется различать понятия рисков традиционного машинного обучения и состязательного обучения ML-модели. С формальной точки зрения риск традиционного (например) классификатора заключается в его ожидаемых потерях при D – истинном

распределении данных в выборке, задаваемых выражением (1):

$$Risk\left(h_\theta\right)=\mathrm{E}_{(\mathbf{x},y)\sim D}\left[\ell\left(M_\theta\left(\mathbf{x}\right),\mathbf{y}\right)\right], \quad (1)$$

где $D=\{x_i,y_i\}$, $i=\overline{1,m}$. $\mathbf{x}$ – вектор входных признаков экземпляра обучающей выборки, $\mathbf{y}$ – вектор целевых истинных меток для заданного класса $\mathbf{x}$, m – мощность набора данных D ($m=|D|$). В выражении (1) $l(\cdot)$ означает функцию потерь ML-модели, функциональность которой обозначена через $M_\theta(\cdot)$, где θ - ее гиперпараметры.

Эмпирический риск соответствует выражению (2):

$$Risk*\left(h_\theta,D\right)=\frac{1}{|D|}\sum_{(\mathbf{x},\mathbf{y})\in D}\ell\left(M_\theta\left(\mathbf{x}\right),\mathbf{y}\right). \quad (2)$$

Тогда традиционный процесс обучения ML-модели заключается в поиске параметров, которые минимизируют эмпирический риск (3) на некотором обучающем наборе данных, обозначенном D_{train}:

$$Risk*\left(M_\theta,D_{train}\right)\rightarrow\min_\theta. \quad (3)$$

В качестве альтернативы традиционному риску рассматривают состязательный риск (4):

$$Risk_{ADV}\left(M_\theta\right)=\mathrm{E}_{(\mathbf{x},y)\sim D}\left[\max_{\delta\in\Delta(x)}\ell\left(M_\theta\left(\mathbf{x}+\delta\right),\mathbf{y}\right)\right], \quad (4)$$

где δ – малое возмущение из множества возмущений $\triangle$.

Существует, естественно, и эмпирический аналог состязательного риска (5):

$$Risk^*_{ADV}\left(h_\theta,D\right)=\frac{1}{|D|}\sum_{(\mathbf{x},\mathbf{y})\in D}\left[\max_{\delta\in\Delta}\ell\left(M_\theta\left(\mathbf{x}+\delta\right),\mathbf{y}\right)\right]. \quad (5)$$

Величина $Risk^*_{ADV}$ измеряет эмпирические потери ML-модели в *наихудшем* случае, когда можно состязательно манипулировать каждым входом в наборе данных D в пределах его допустимого набора возмущений $\triangle$.

Известно множество методов для генерации состязательных примеров, таких как генерация шумовых гауссовских примеров [13], синтез состязательных примеров генеративными моделями (в частности, диффузионными) [15], градиентные методы искажения параметров моделей [10-12]. В контексте оценки устойчивости к состязательности YOLO-детекторов целесообразным является рассмотрение состязательных атак, способных быть выполненными в реальном (или близком к реальному) масштабе времени. В силу этого безусловно перспективные диффузионные модели, используемые для реализации так называемых CAP-атак [15], и требующие на сегодняшний день для своей реализации значительных временных и вычислительных ресурсов, в данной работе не рассматриваются. С шумовыми состязательными примерами $\mathbf{x'}=\mathbf{x}+\epsilon$, $\epsilon\sim N\left(0,\sigma^2\right)$, где ϵ – аддитивный шум, полученный из нормального распределения с нулевым средним значением и дисперсией σ^2, современные версии YOLO-детекторов способны эффективно бороться за счет внутренних механизмов шумоподавления [1]. Поэтому наиболее широко применяемым подходом является градиентный спуск, при этом общая стратегия состязательного обучения ML-модели M_θ заключается в следующем:

Шаг 1. Выбрать мини-пакет B набора данных D и инициализировать вектор градиентов $\mathbf{g}$: = 0.

Шаг 2. Для каждого экземпляра (x, y) из B выполнить:

1) Найти возмущение градиента δ⋆ путем (приблизительной) оптимизации

$$*=\qquad\qquad\ell\left(\quad_\theta\left(\quad+\right.\right.$$
$$\|\delta\|\leq\varepsilon$$

2) Добавить δ⋆ в градиент ML-модели:

$$g:=g+\nabla_\theta\ell\left(M_\theta\left(x+\delta*\right),y\right).$$

Шаг 3. Обновить параметры ML-модели M_θ.

Когда YOLO-модель работает в состязательной среде, где злоумышленник способен манипулировать входными данными, полностью зная ее архитектуру (атака типа white box), то именно использование стратегии состязательного обучения позволяет обеспечить снижение деградации производительности при выполнении black-box состязательных атак, а также более точную оценку ожидаемой производительности ML-модели в целом. Как будет показано далее, нативные YOLO-модели (не обученные в состязательной среде) достаточно хрупкие в контексте состязательности, и состязательные примеры обнажают этот факт очень очевидным и интуитивно понятным образом.

Оптимизационную задачу состязательного обучения формально можно представить в виде выражения (6):

$$\min_\theta Risk^*_{ADV}\equiv\min_\theta\frac{1}{|D_{train}|}\sum_{(\mathbf{x},\mathbf{y})\in D_{train}}\left[\max_{\delta\in\Delta}\ell\left(M_\theta\left(\mathbf{x}+\delta\right),\mathbf{y}\right)\right] \quad (6)$$

Минимаксную задачу, соответствующую формулировке (6), называют надежной (robust - робастной) оптимизацией состязательного обучения (robust optimization of adversarial learning). Как и в случае с традиционным обучением, на практике задачу оптимизации (6) обычно решают методом стохастического градиентного спуска с опорой на теорему Данскина (Danskin's theorem) [13]. Здесь следует отметить, что каждая состязательная атака и защита являются методом *приближенного* решения проблемы внутренней максимизации и/или внешней минимизации соответственно. Тщательное решение задачи внутренней оптимизации позволяет создать стратегии сильных атак и надежных механизмов защиты, но это не гарантирует наличие абсолютной защиты или универсальной успешной атаки.

Поскольку в данной работе сделан акцент на состязательные атаки, направленные на градиенты YOLO-детекторов, следует проанализировать их функции потерь. В ходе своей эволюции в YOLO-моделях использовались разные функции потерь (табл.1). Большинство современных версий моделей семейства YOLO обладают комплексной функцией потерь (7), включающей в свой состав, как правило, три терма – $l_{Regression}$ (8), оценивающий регрессионные потери ограничивающих рамок b-box (терм локализации), $l_{Objectness}$ (9) оценивающий уровень уверенности обнаружения объекта (терм уверенности модели в том, что внутри b-box находится центр

объекта), и и терм $l_{Classification}$ (10), оценивающий кросс-энтропию классификации обнаруженного объекта (терм классификации).

$$l_{YOLO} = l_{Regression} + l_{Objectness} + l_{Classification}. \quad (7)$$

$$l_{Regression} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(c_i^x - \hat{c}_i^x \right)^2 + \left(c_i^y - \hat{c}_i^y \right)^2 \right] + \quad (8)$$

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right],$$

где S – размер сетки, разделяющих входное изображение на ячейки, B – число предсказанных b-boxes, 1_{ij}^{obj} – индикаторная функция (равна 1, если объект обнаружен и равна 0 в противном случае, λ_{coord} – коэффициент регуляризации, $c_i^x, \hat{c}_i^x$ – абсциссы центров b-box предсказанного и истинного соответственно, $c_i^y, \hat{c}_i^y$ – ординаты центров b-box предсказанного и истинного соответственно, $w_i, \hat{w}_i$ – значения ширины, а $h_i, \hat{h}_i$ – значения высоты b-box предсказанного и истинного i-го объекта соответственно.

$$l_{Objectness} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log(1 - C_i) \right) + \quad (9)$$

$$+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log(1 - C_i) \right),$$

где $C_i, \hat{C}_i$ – IoU с нормализацией для истинного и прогнозируемого b-box i-го объекта. Этот терм учитывает не только те ячейки, в которых есть объект, но и те, в которых нет объектов (*noobj*) – на них модель обучается давать меньше ложноположительных ответов. Такие ячейки регулируются коэффициентом λ_{noobj}, который обычно примерно в 10 раз меньше λ_{coord}.

$$l_{Classification} = \sum_{i=0}^{S^2} \sum_{k \in classes} 1_{ij}^{obj} \left(\hat{p}_i(k) \log\left(p_i(k)\right) + (1 - \hat{p}_i(k)) \log\left(1 - p_i(k)\right) \right), \quad (10)$$

$p_i(k), \hat{p}_i(k)$ – вероятности присвоения i-му объекту метки k-го класса для прогнозируемого и истинного объектов соответственно.

Таблица 1

Используемые функции потерь в YOLO-моделях

Модель	Используемая функция потерь
YOLOv1	MSE для регрессии b-box
YOLOv1	MSE для регрессии b-box Cross-Entropy для классификации
YOLOv5, YOLOv4, YOLOR и YOLOv7	L_CIoU = 1-Complete IoU= 1- (IoU – (C \(A U B))/C)
YOLOX	L_IoU = 1 - IoU
YOLOv6	L_SIoU/GIoU для регрессии b-box (L_GIoU=1 - GIoU = 1 - IoU + (C \(A U B))/C) VariFocal для задачи классификации
YOLOv8	L_CIoU и Distribution Focal Loss (DFL)
YOLOv11и выше	Выражение (7)

При состязательной атаке на YOLO-модель могут быть сгенерированы состязательные примеры, являющиеся решением минимаксной задачи состязательного риска на градиенты каждого из термов l_{YOLO} (7) как в отдельности, так и совместно. При этом атака на градиенты терма локализации (8) приводит к несанкционированному смещению либо центра b-box, либо его размеров. При атаке на градиенты терма уверенности (9) фиксируется несанкционированное появление (или непоявление) b-box, а при атаке на градиенты терма классификации производится целевой (или нецелевой) переворот метки класса. Причем атакованы градиенты могут быть как посредством применения состязательных примеров, так и непосредственной модификацией градиентов YOLO-модели (например, при атаках на цепочки поставок ML-модели при ее развертывании). В последнем случае модифицируются весовые параметры различных слоев YOLO – как сверточных, так и полносвязных.

Производительность детекторов объектов обычно оценивается с помощью таких метрик, как Intersection over Union (IoU) и mean Average Precision (mAP). Метрика mAP, являясь интегральной, эффективно балансирует между нахождением всех релевантных объектов (Recall) и гарантией того, что найденные объекты действительно правильные (Precision). То есть она одновременно учитывает и точность классификации (правильно ли определен класс объекта?), и точность локализации (правильно ли размещена ограничительная рамка?) для всех классов, определенных в обучающих данных.

В данной работе рассчитывались значения mAP@0.5-0.95 (или mAPval 50-95 в процентном выражении) путем усреднения mAP по нескольким пороговым значениям IoU от 0,5 до 0,95 с шагом 0,05. Следует отметить, что вычисление mAP начинается с сортировки по степени уверенности для каждого класса объектов предсказаний модели (обнаруженные b-box) от самого высокого к самому низкому. Следовательно, атаки, направленные на градиенты терма уверенности функции потерь (9), могут быть наиболее эффективными и существенно снизить mAP, что и экспериментально подтверждено в ходе данного исследования.

Без потери общности будем в качестве базовой атаки, направленной на градиенты, использовать атаку PGP (Projected Gradient Descent) [13]. Тогда, учитывая обозначения, используемые в выражениях (8) – (10), в формальном виде итеративное формирование состязательных примеров для YOLO-модели можно описать следующим образом:

$$\mathbf{x}_{Obj}^{(i+1)} = \mathbf{Proj}_{x+\epsilon}\left(\mathbf{x}_{Obj}^{(i)} + \alpha \cdot g \right), \quad \text{for } g = sign\left(\nabla_{\mathbf{x}_{Obj}^{(i)}} l_{Objectness}\left(\mathbf{x}_{Obj}^{(i)}, B; \theta \right) \right), \quad (11)$$

$$\mathbf{x}_{Reg}^{(i+1)} = \mathbf{Proj}_{x+\epsilon}\left(\mathbf{x}_{Reg}^{(i)} + \alpha \cdot g \right), \quad \text{for } g = sign\left(\nabla_{\mathbf{x}_{Reg}^{(i)}} l_{Regresion}\left(\mathbf{x}_{Reg}^{(i)}, B; \theta \right) \right),$$

$$\mathbf{x}_{Class}^{(i+1)} = \mathbf{Proj}_{x+\epsilon}\left(\mathbf{x}_{Class}^{(i)} + \alpha \cdot g \right), \quad \text{for } g = sign\left(\nabla_{\mathbf{x}_{Class}^{(i)}} l_{Classification}\left(\mathbf{x}_{Class}^{(i)}, B; \theta \right) \right),$$

где функция **Proj** ( ) проецирует входные данные **x** на меньшую область (гипершар I с центром в точке **x**) с незначительными искажениями, α – размер шага атаки, $\epsilon > 0$ – сила (бюджет) атаки. Здесь бюджет атаки определяет максимальное возмущение в терминах l_∞ – нормы.

Атака начинается в случайной точке $\mathbf{x}^{(0)} \in B$, $\mathbf{x}^{(0)}_{\text{Obj}} = \mathbf{x}^{(0)}_{\text{Reg}} = \mathbf{x}^{(0)}_{\text{Class}} = \mathbf{x}^{(0)}$ и многократно выполняется (11). После N итера-

ционных шагов предполагается, что $\mathbf{x}^{(N)}$ становится результирующим состязательным примером. Если N=1, то PGP атака становится атакой FGSM (Fast Gradient Sign Method) [13]. В силу того, что при генерации состязательных примеров FGSM требует только одной оценки градиента, подобная атака вычислительно эффективна. Кроме того, FGSM и PGP применимы непосредственно к пакету входных данных, что позволяет генерировать большое число состязательных примеров.

Кратко говоря об архитектуре YOLO-детекторов [1], отметим, что исследуется чувствительность основных компонент YOLO-модели («позвоночник» – backbone, «шею» – neck и «головы» – heads) к несанкционированной модификации их весовых параметров. То есть замораживание слоев в ходе экспериментов не производилось. Backbone отвечает за извлечение признаков и формирует карту признаков (feature map). В YOLOv11 позвоночник использует модифицированную архитектуру CSP (Cross Stage Partial), которая улучшает изучение признаков, одновременно снижая вычислительные затраты. Neck объединяет (аггрегирует) особенности признаков из разных слоев, чтобы улучшить способность модели обнаруживать объекты в разных масштабах. Для этой цели YOLOv11 использует PANet (Path Aggregation Network), улучшая обнаружение объектов разных масштабов и ориентаций. Head формирует окончательный вывод: для задачи обнаружения объектов – ограничивающие рамки (b-boxes), метки классов и оценки уверенности (confidence scores).

Следует отметить, что YOLOv11 спроектирована в 2024 году компанией Ultralitics [1] и достигает повышенной точности обнаружения объектов за счет использования сложных функций потерь (1.7), улучшенных якорных b-boxes и улучшенного механизма аугментации данных. Она подходит для приложений, требующих мгновенной обратной связи, таких как анализ «живого» видео.

2 Стратегия комплексирования при генерации состязательных примеров

Существует множество различных состязательных атак, которые можно использовать против систем машинного обучения. В соответствии с классификацией NIST на сегодняшний день данные атаки подразделяют на четыре большие категории: отравляющие атаки (Data Poisoning и Model Poisoning); атаки уклонения (Evasion); атаки извлечения модели (Model Extraction): атаки злоупотребления генеративным искусственным интеллектом (Abuse).

Детекторы YOLO относятся к классу предиктивных моделей, и в данной работе акцент сделан на атаки отравления и атаки уклонения. Напомним, что эксплуатация атак уклонения относится к типу «black box» и проводится на этапах логического вывода (Model Inference), используя искусно подобранные состязательные примеры на входе модели, а эксплуатация атак отравления проводится на этапах обучения (в нашем случае, трансферного) и развертывания модели при несанкционированных модификациях датасетов и/или параметров модели.

В ходе данного исследования проводился ряд состязательных атак отравления, базирующихся на градиентных стратегиях (типа PGP). В отличии от существующих имплементаций атак на YOLO-детекторы, ориентированных на атаки либо на терм классификации, либо на терм локализации объектов, в данном исследовании учитывалось различное комплексирование состязательных модификаций исходных образцов датасетов в соответствии с выражениями (11). Для этого в целевую функцию потерь (7) введены коэффициенты регуляризации γ_1, γ_2 , γ_3 (соответственно для терма локализации, уверенности и классификации).

$$l_{YOLO} = \gamma_1 \cdot l_{Regression} + \gamma_2 \cdot l_{Objectness} + \gamma_3 \cdot l_{Classification}. \quad (12)$$

Нулевые значения коэффициентов регуляризации γ позволяют отключать тот или иной терм при генерировании состязательных примеров.

Предлагаемый алгоритм комплексирования состязательных модификаций (COMPLEX_ADV) с последующим состязательным обучением может быть описан следующим образом:

Алгоритм COMPLEX_ADV.

Шаг 1. Инициализация: Датасет – D, параметры YOLO-модели – θ, размер пакета – B, количество эпох обучения – EPOCHS, число вокеров – WORKERS, бюджет атаки $\epsilon > 0$, размер шага атаки α=1. Коэффициенты регуляции термов функции потерь – $\gamma_1, \gamma_2, \gamma_3$.

Шаг 2. Генерация комплексного пакета состязательных примеров и состязательное обучение

for j=1 to EPOCHS do

for Random k in BATCH $\left\{\mathbf{x}^i, \left\{y^i, \mathbf{b}^i\right\}\right\}_{k=1}^{B} \sim D$ do

for i=1 in ϵ

$$\mathbf{x}_{Obj}^{(i+1),k} = \mathbf{Proj}_{x+\epsilon}\left(\mathbf{x}_{Obj}^{(i),k} + \alpha \cdot g\right), \quad g = sign\left(\nabla_{\mathbf{x}_{Obj}^{(i)}} l_{Objectness}\left(\mathbf{x}_{Obj}^{(i),k}, B; \theta\right)\right),$$

$$\mathbf{x}_{Reg}^{(i+1),k} = \mathbf{Proj}_{x+\epsilon}\left(\mathbf{x}_{Reg}^{(i),k} + \alpha \cdot g\right), \quad g = sign\left(\nabla_{\mathbf{x}_{Reg}^{(i)}} l_{\text{Regression}}\left(\mathbf{x}_{Reg}^{(i),k}, B; \theta\right)\right),$$

$$\mathbf{x}_{Class}^{(i+1),k} = \mathbf{Proj}_{x+\epsilon}\left(\mathbf{x}_{Class}^{(i),k} + \alpha \cdot g\right), \quad g = sign\left(\nabla_{\mathbf{x}_{Class}^{(i)}} l_{Classification}\left(\mathbf{x}_{Class}^{(i),k}, B; \theta\right)\right),$$

Выделить и сохранить пакет состязательных примеров:

$$\underline{\mathbf{x}}^{(i+1),k} = \arg \max_{\overline{\mathbf{x}}^{(i+1),k} \in \left\{\mathbf{x}_{Obj}^{(i+1),k}, \mathbf{x}_{Reg}^{(i+1),k}, \mathbf{x}_{Class}^{(i+1),k}\right\}} L_{YOLO}\left(\overline{\mathbf{x}}^{(i+1),k}, \left\{y^{(i+1),k}, \mathbf{b}^{(i+1),k}\right\}; \theta\right)$$

для L_{YOLO}, задаваемой выражением (12)

$$l_{YOLO} = \gamma_1 \cdot l_{Regression} + \gamma_2 \cdot l_{Objectness} + \gamma_3 \cdot l_{Classification}.$$

При необходимости, зафиксировать деградацию метрики производительности

Состязательное обучение:

$$\theta' = \arg\min_{\theta} L_{YOLO}\left(\mathbf{x}^{(i),k}, \left\{y^{(i),k}\mathbf{b}^{(i),k}\right\}; \theta\right) + L_{YOLO}\left(\overline{\mathbf{x}}^{(i+1),k}, \left\{y^{(i+1),k}\mathbf{b}^{(i+1),k}\right\}; \theta\right)$$

End for i

End for k

End for j

Шаг 4. Сохранить обученную YOLO-модель с параметрами θ'.

На шаге 2 из всего спектра существующих решений (AdvBox, Deepsec, Cleverhans, Foolbox и др.), в качестве инструмента имплементации состязательных атак PGP и FGSM использована функциональность библиотеки ATR (Adversarial Robustness Toolbox), как наиболее полной по набору модулей атак и актуализированная на текущий момент сообществом разработчиков.

Данная библиотека помимо механизмов атак предоставляет и механизмы оценки производительности и защиты ML-моделей, в частности, такие как состязательное обучение, а также сертификация и верификация модели.

3 Экспериментальное трансферное обучение YOLO-детекторов

Экспериментальные исследования проводились в следующей последовательности:

1) Трансферное обучение YOLO-детекторов на датасетах со специализацией классов в транспортной сфере.

2) Генерация состязательных примеров и использование градиентных стратегий атак, как на градиенты отдельных термов функции потерь (12), так и на их комплексное использование в соответствии с авторским алгоритмом в соответствии с алгоритмом.

3) Состязательное обучение YOLO-детекторов.

4) Сравнительная оценка метрик качества YOLO-детекторов в ходе атак на YOLO-детекторы, не обученные и обученные в состязательной среде.

Для сопоставимости результатов проведенных экспериментов использовались детекторы серии medium YOLOv5, YOLOv8, YOLOv9, YOLOv10 и YOLOv11, предварительно обученные на одном и том же известном и распространенном наборе данных Traffic COCO (использовалась версия v3 от 2023-12-09 с изображениями размерностью 640x640, аннотированных по 15 классам [https://universe.roboflow.com/ailker7/cocotraffic/dataset/3]. Traffic COCO является подмножеством датасета COCO (Common Objects in Context), специализирующемся на классах, являющихся транспортными средствами и участниками дорожного движения. Класс «светофор» (индекс 10 в наборе данных COCO) в Traffic COCO разделен на три отдельных класса: «traffic_light_red», «traffic_light_green» и «traffic_light_na». Эти новые классы позволяют модели определять состояние светофора и решать, может ли транспортное средство безопасно проехать через перекресток.

В ходе дальнейшего трансферного обучения, используемые в экспериментах YOLO-детекторы дообучались на двух разных (также общедоступных) наборах данных, а именно на наборе данных Udacity Self-driving Car (USC) [https://public.roboflow.com/object-detection/self-driving-car] и наборе данных Vehicles-OpenImages (VO).

Набор данных Vehicles-OpenImages содержит всего 627 изображений различных классов транспортных средств, подготовленных для решения задач обнаружения объектов, таких как «Car», «Bus», «Ambulance», «Motorcycle» и «Truck». На 627 изображениях имеется 1194 области интереса (объекта), это означает, что на изображение приходится не менее 1,9 объектов.

Согласно эвристике, показанной на рисунке 1, класс «Car» составляет более 50% объектов. Напротив, остальные классы представлены недостаточно по сравнению с классом «Car».

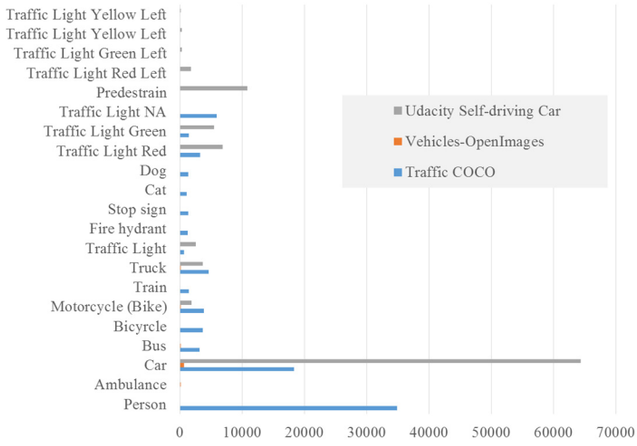


Рис. 1. Распределение классов в используемых наборах данных

Используемый в работе набор данных Udacity для беспилотных автомобилей получен из оригинального набора данных USC, включает изображения с трех камер автомобиля (левой, центральной и правой). В оригинальном наборе данных отсутствовали метки для тысяч объектов, поэтому использовался перемаркированный компанией Roboflow вариант этого датасета. Он содержит 15 000 изображений, разбитых по 11 классам с 97 942 метками.

Также потребовалась *crop*-преобразование изображений в размерность 640x640 с частичной потерей краевых объектов. Как и набор данных Vehicles-OpenImages, этот датасет не сбалансирован – содержит наибольшее количество объектов, относящихся к классу «Car» (66 % от общего числа объектов). В итоге формировался объединенный набор данных $D = \{TrafficCOCO'\} \cup \{VO'\} \cup \{USC'\}$ с 5 релевантными категориями объектов, представленными во всех используемых датасетах.

Решалась задача обнаружения объектов. Для всех YOLO-детекторов использовались одинаковые параметры обучения BATCH_SIZE = 32, EPOCHS = 100, WORKERS = 1, оптимизатор – Adam с начальной скоростью обучения 10^{-3}. На чистых образцах получены значения метрик mAP, Precision, Recall, F1-SCORE, отображенные в таблице 2.

Трансфертное обучение позволило слегка улучшить значения метрик mAP по сравнению с значениями, полученными при обучении на полном датасете COCO. Для сравнения на рисунке 2 приведен график метрик mAP для разных серий YOLOv11, полученных на полном датасете COCO.

Таблица 1

Исходные примеры изображений и результаты обнаружения объектов, полученные после трансферного обучения YOLOv11m

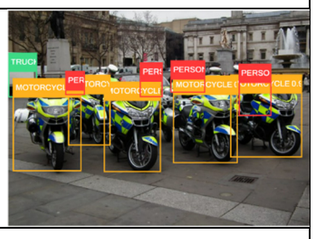
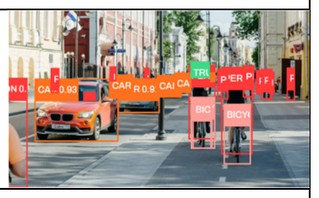
Экземпляр из валидационной выборки (набор данных)	Обнаруженные объекты по результатам трансферного обучения YOLOv11m
	
(Traffic COCO)	
	
(Traffic COCO)	
	
(Инференс-экземпляр)	

Таблица 2

Значения метрик качества YOLO-моделей, полученные на чистых (не состязательных) примерах после трансферного обучения

Версия YOLO-модели	mAP 50-95, %	Precision	Recall	F1-Score
Набор данных Vehicles-OpenImages (VO)				
YOLOv5	54	63	62,7	62,8
YOLOv8	54,8	72,1	68,2	70,1
YOLOv9	46,5	56,3	54,2	55,2
YOLOv10	47,9	77,3	56,4	65,8
YOLOv11	55,6	66,2	65,6	65,9
Набор данных Udacity Self-driving Car (USC)				
YOLOv5	54,5	64,5	62,7	63,6
YOLOv8	55,4	71,1	69,1	70,1
YOLOv9	44,1	55,3	51,2	53,2
YOLOv10	48,2	79,1	56,4	65,8
YOLOv11	56,4	66,7	65,3	66,0

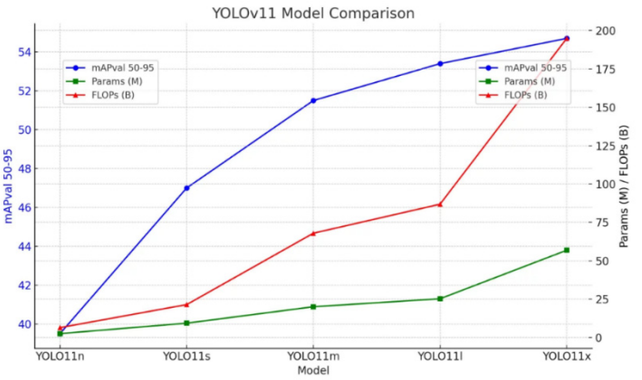


Рис. 2. Значения характеристик YOLOv11, обученных на полном наборе данных COCO
Источник: https://github.com/ultralytics/ultralytics/issues/17102

4 Комплексная состязательная атака и состязательное дообучение YOLO-детекторов

Комплексная состязательная атака с последующим состязательным обучением, как мерой защиты, проводилась в соответствии с предложенным алгоритмом COMPLEX_ADV над набором данных $D = \{TrafficCOCO'\} \cup \{VO'\} \cup \{USC'\}$, разделенным на обучающие, тестовые и валидационные выборки – 70/20/10%, причем в каждую из подвыборок случайно добавлялись по 25% состязательных примеров. При состязательном обучении YOLO-детекторы переобучались на соответствующих состязательных примерах.

Параметры обучения YOLO-модели устанавливались идентичными трансферному обучению, бюджет атаки $\epsilon \in \{0,2,4,8,16,32\}$, где нулевое значение соответствует отсутствию атаки. Размер шага атаки α=1. Коэффициенты регуляции термов функции потерь – γ_1=[0;1], γ_2=[0;1] и γ_3=[0;1]. Количество итераций – для FSGM T=1, для PGD – T=10.

Результаты, полученные в ходе экспериментов отражены в табл. 3 для атак, направленных на терм локализации, в табл. 4 – на терм классификации, в табл. 5 – на терм уверенности, и в табл. 6 представлена деградация показателей mAPval-0,95 для комплексной атаки одновременно на все термы. Следует отметить, что ожидаемо атаки на терм уверенности являются наиболее сильными и приводят к резкой деградации mAPval при выполнении PGD-атак.

Таблица 3

Деградация mAPval 50-95 (%) при атаках на терм локализации

Версия YOLO-модели	Тип атаки /Бюджет атаки	0	2	4	8	16	32
YOLOv5	FGSM до AT	54,5	53,83	52,32	50,36	47,40	38,37
	FGSM после AT	54,5	54,45	53,96	53,68	53,63	53,15
	PGD до AT	54,5	53,67	51,70	49,74	47,02	32,25
	PGD после AT	54,5	54,45	54,39	54,34	54,28	54,23
YOLOv8	FGSM до AT	55,4	55,34	54,85	54,57	54,29	54,24
	FGSM после AT	55,4	55,34	54,85	54,57	54,51	54,02
	PGD до AT	55,4	54,56	51,76	47,23	40,75	24,11
	PGD после AT	55,4	55,34	55,29	55,23	55,18	55,12

YOLOv9	FGSM до AT	44,1	44,06	43,66	43,44	43,22	43,17
	FGSM после AT	44,1	44,06	43,66	43,44	43,40	43,00
	PGD до AT	44,1	43,43	41,20	37,60	32,44	19,19
	PGD после AT	44,1	44,06	44,01	43,97	43,92	43,88
YOLOv10	FGSM до AT	48,2	48,15	47,72	47,48	47,24	47,19
	FGSM после AT	48,2	48,15	47,72	47,48	47,43	47,00
	PGD до AT	48,2	47,47	45,03	41,10	35,45	20,98
	PGD после AT	48,2	48,15	48,10	48,06	48,01	47,96
YOLOv11	FGSM до AT	56,4	56,34	55,84	55,55	55,27	55,22
	FGSM после AT	56,4	56,34	55,84	55,55	55,50	55,00
	PGD до AT	56,4	55,54	52,69	48,09	41,48	24,55
	PGD после AT	56,4	56,34	56,29	56,23	56,17	56,12

Таблица 4

Деградация mAP (%) при атаках на терм классификации

Версия YOLO-модели	Тип атаки /Бюджет атаки	0	2	4	8	16	32
YOLOv5	FGSM до AT	54,5	53,84	52,36	50,66	47,93	41,78
	FGSM после AT	54,5	54,45	53,96	53,68	53,63	53,15
	PGD до AT	54,5	53,91	52,70	50,47	47,97	45,86
	PGD после AT	54,5	54,45	54,39	54,34	54,28	54,23
YOLOv8	FGSM до AT	55,4	55,34	54,85	54,57	54,29	54,24
	FGSM после AT	55,4	55,34	54,85	54,57	54,51	54,02
	PGD до AT	55,4	54,80	52,99	49,08	43,20	36,35
	PGD после AT	55,4	55,34	55,29	55,23	55,18	55,12
YOLOv9	FGSM до AT	44,1	44,06	43,66	43,44	43,22	43,17
	FGSM после AT	44,1	44,06	43,66	43,44	43,40	43,00
	PGD до AT	44,1	43,62	42,18	39,07	34,39	28,94
	PGD после AT	44,1	44,06	44,01	43,97	43,92	43,88
YOLOv10	FGSM до AT	48,2	48,15	47,72	47,48	47,24	47,19
	FGSM после AT	48,2	48,15	47,72	47,48	47,43	47,00
	PGD до AT	48,2	47,68	46,11	42,70	37,59	31,63
	PGD после AT	48,2	48,15	48,10	48,06	48,01	47,96
YOLOv11	FGSM до AT	56,4	56,34	55,84	55,55	55,27	55,22
	FGSM после AT	56,4	56,34	55,84	55,55	55,50	55,00
	PGD до AT	56,4	55,79	53,95	49,96	43,98	37,01
	PGD после AT	56,4	56,34	56,29	56,23	56,17	56,12

Как видно из таблиц 3-5, атаки, направленные на градиенты только термов локализации и классификации, приводят к существенно меньшей деградации общей производительности YOLO-детекторов. Основные ошибки классификации и смещения параметров b-boxes, возникшие при состязательных атаках, фиксируются на отдаленных объектах изображений. Это связано с разреженностью и низкой детализацией удаленных пикселей обнаруживаемых объектов, в которых случайные преобразования, как правило, искажают или полностью стирают ключевые признаковые особенности.

Следовательно, состязательное обучение при обобщении изменяет характеристики доменного пространства, оптимизируя надежность локализации и классификации объектов в ближнем поле зрения, но ограничивая обобщение за пределами этого диапазона. Возможно, при проведении многоступенчатого состязательного обучения (что не производится в рамках алгоритма COMPLEX_ADV), при котором генераторы состязательных примером будут оценивать удаленность объекта, удастся уменьшить уровень деградации производительности моделей.

Таблица 5

Деградация mAP (%) при атаках на терм уверенности

Версия YOLO-модели	Тип атаки / Бюджет атаки	0	2	4	8	16	32
YOLOv5	FGSM до AT	54,5	52,1	50,4	44,1	37,8	30,6
	FGSM после AT	54,5	54,0	52,1	50,7	49,9	48,1
	PGD до AT	54,5	50,6	25,5	7,6	2,3	0,2
	PGD после AT	54,5	53,4	49,4	44,3	34,3	30,2
YOLOv8	FGSM до AT	55,4	53,3	50,8	46,6	40,3	34,2
	FGSM после AT	55,4	55,0	52,8	51,6	50,5	49,0
	PGD до AT	55,4	51,0	19,3	3,0	0,8	0,2
	PGD после AT	55,4	54,3	50,3	45,2	37,2	35,1
YOLOv9	FGSM до AT	44,1	43,7	41,4	38,4	34,2	30,2
	FGSM после AT	44,1	44,1	43,0	42,4	40,4	39,0
	PGD до AT	44,1	40,7	17,4	2,4	0,6	0,2
	PGD после AT	44,1	42,1	40,0	36,0	30,9	27,9
YOLOv10	FGSM до AT	48,2	46,2	43,7	40,5	37,2	35,2
	FGSM после AT	48,2	48,1	46,7	45,5	44,4	41,0
	PGD до AT	48,2	42,7	18,1	2,6	0,7	0,2
	PGD после AT	48,2	47,2	44,1	40,1	32,0	28,0
YOLOv11	FGSM до AT	56,4	55,3	54,8	53,6	51,3	48,2
	FGSM после AT	56,4	56,3	55,0	54,6	53,3	50,2
	PGD до AT	56,4	49,5	29,5	3,1	0,8	0,2
	PGD после AT	56,4	55,3	54,3	50,2	46,2	44,1

Таблица 6

Деградация mAPval 50-95 (%) при комплексных атаках

Версия YOLO-модели	Тип атаки / Бюджет атаки	0	2	4	8	16	32
YOLOv5	FGSM до AT	54,5	53,8	51,1	47,5	39,1	31,6
	FGSM после AT	54,5	54,4	54,0	53,7	53,6	53,1
	PGD до AT	54,5	50,6	25,5	7,6	2,3	0,2
	PGD после AT	54,5	53,4	49,4	44,3	34,3	30,2
YOLOv8	FGSM до AT	55,4	54,3	52,8	50,6	45,3	34,2
	FGSM после AT	55,4	55,3	54,8	54,6	54,5	54,0
	PGD до AT	55,4	51,0	19,3	3,0	0,8	0,2
	PGD после AT	55,4	54,3	50,3	45,2	37,2	35,1
YOLOv9	FGSM до AT	44,1	44,1	43,7	41,4	38,2	34,2
	FGSM после AT	44,1	44,1	43,7	43,4	43,4	43,0
	PGD до AT	44,1	40,7	17,4	2,4	0,6	0,2
	PGD после AT	44,1	42,1	40,0	36,0	30,9	27,9
YOLOv10	FGSM до AT	48,2	48,2	45,7	42,5	39,2	35,2
	FGSM после AT	48,2	48,2	47,7	47,5	47,4	47,0
	PGD до AT	48,2	42,7	18,1	2,6	0,7	0,2
	PGD после AT	48,2	47,2	44,1	40,1	32,0	28,0
YOLOv11	FGSM до AT	56,4	56,3	55,8	53,6	51,3	48,2
	FGSM после AT	56,4	56,3	55,8	55,6	55,3	55,2
	PGD до AT	56,4	49,5	29,5	3,1	0,8	0,2
	PGD после AT	56,4	55,3	54,3	50,2	46,2	44,1

Примеры обнаружения объектов до и после комплексной атаки представлены на рис. 3. При наличии на исходном изображении 6 объектов класса «Person», 6 «Motorcycle» и 1 объекта класса «Truck», состязательный пример, поданный на вход не обученной состязательности YOLOv11m при бюджете PGP атаки равном 16 приводит к появлению большого числа ложноположительных срабатываний – 10 «Person», 8 «Motorcycle» и 1 «Truck». График деградации при FGSM-атаках представлен на рисунке 4.

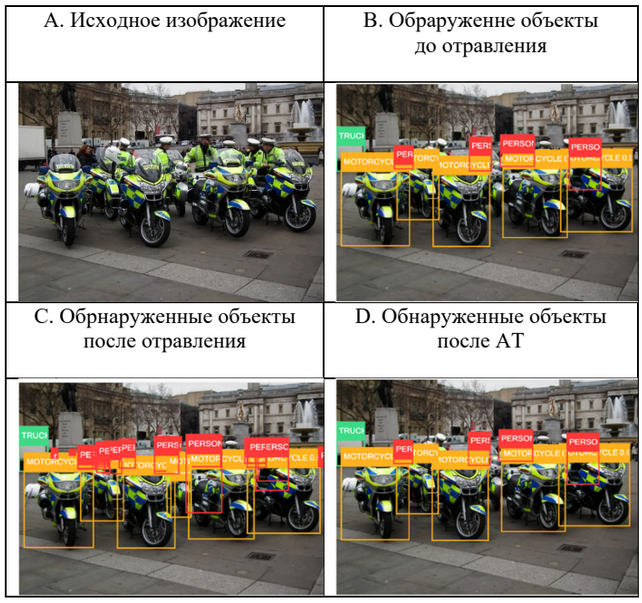


Рис. 3. Результаты обнаружения объектов в разных инференс-режимах

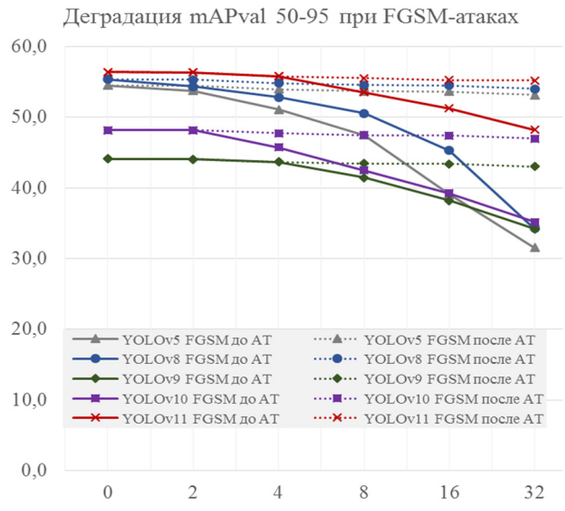


Рис. 4. Деградация производительности YOLO-детекторов до и после состязательного обучения (AT) при комплексных FSGM-атаках

После добавления пакета сгенерированных состязательных примеров в обучающую выборку и переобучения модели для данного примера обнаружение объектов выполнено уже корректно. Однако полноценное восстановление изначальной производительности, как следует это из рисунка 5, не достигается, значения mAPval 50-95 снижаются на 30% и более для атак с большим бюджетом. Это подтверждает факт того, что итеративные градиентные атаки остаются сильными.

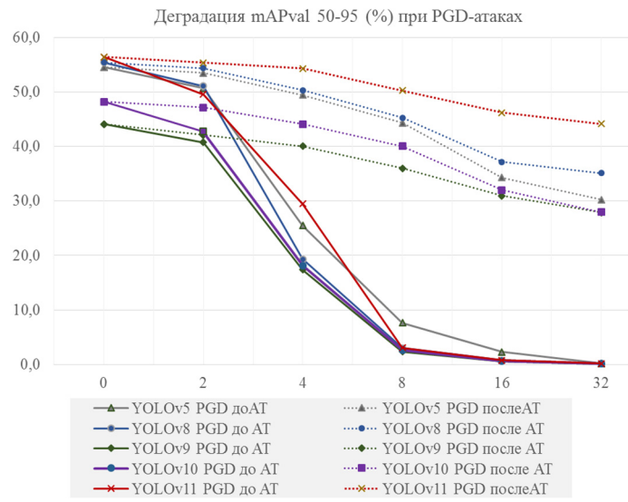


Рис. 5. Деградация производительности YOLO-детекторов до и после состязательного обучения (AT) при комплексных PGP-атаках

Сравнивая устойчивость к состязательным атакам всего YOLO-семейства, следует отметить, что несмотря на одинаковые тенденции деградации, при малых бюджетах атак YOLOv11m за счет имеющихся механизмов шумоподавления, в целом удерживает показатели производительности на более высоких уровнях, но глубокие градиентные атаки остаются эффективными. Для их дальнейшего смягчения требуются дополнительные меры защиты, такие как

1) Использование доверенных сред выполнения, где обучение и эксплуатация предварительно верифицированной модели п ее наборов данных производится в изолированной среде [16, 17]. Данный подход требует реализации механизмов удаленной аттестации программных и аппаратных сущностей, участвующих в трансферном обучении.

2) Методы ансамблирования моделей, где за счет снесения архитектурной избыточности, ансамбль моделей выносит более надежное мажоритарное или компромиссное решение. Данные методы требуют дополнительных вычислительных ресурсов, но, как правило, обеспечивают повышенную производительность в случаях состязательных атак [13].

3) Методы предварительной обработки изображений перед их вводом в YOLO-детектор. Предварительная обработка входных образов может быть выполнена заранее, тем самым предоставляя YOLO-детектором в реальном времени выполнять свою основную задачу. Однако, базовым недостатком таких подходов является возможное снижение производительности модели при отсутствии атак, причем иногда более существенное, чем после состязательного обучения). Среди основных методов предварительной обработки изображений следует выделить:

- *медианное размытие* [13, 18], где для подавления состязательных модификаций, используется медианная фильтрация, заменяющая каждый пиксель изображения медианой его окрестности, что эффективно сохраняет границы и уменьшает возмущения.

- *уменьшение разрядности* [13, 18], при котором осознанно снижают точность значений, отсекая младшие значащие биты в значениях пикселях, тем самым снижая эффективность малозаметных воздействий, размещенных в наименее значимых битах.
- *рандомизация* [13, 19], при которой преднамеренно привносят случайности, такие как случайное изменение размера, заполнение или введение шума во время предварительной обработки, что приводит к нарушению воздействий злоумышленников, в частности, при реализации атак уклонения

4) Методы состязательного обучения [13], в том числе непосредственно в ходе трансферного и/или итеративного обучения.

5) Котрастивное обучение [13, 20], в ходе которого модели обучаются сближать расширенные представления одного и того же экземпляра в пространстве внедрения (эмбандинга), одновременно разделяя несвязанные экземпляры, часто с помощью оценки контрастности шума (noise contrastive estimation- NCE). Акцент контрастивного обучения на инвариантности признаков снижает чувствительность к вариациям входных данных позволяет обеспечить некоторую устойчивость к простым возмущениям, основанным на градиенте, таким как FGSM, но не гарантирует защиту от глубоких PGD-атак.

6) Методы смягчения несанкционированных модификаций, внесенных в ходе состязательных атак, например, используя восстановление поврежденных изображений и их разметки с помощью диффузионных моделей [15]. Следует отметить, что для восстановления искаженных объектов, не находящихся в ближнем поле зрения, данный подход слабо продуктивен, поскольку на больших расстояниях изображения, восстановленные методом диффузии, часто приводят к ложноотрицательным прогнозам, при этом деградация производительности моделей усиливается в отсутствии подобных атак.

Заключение

В данной работе приведены результаты исследования устойчивости интеллектуальных детекторов объектов семейства YOLO к состязательным атакам, направленным на градиенты через отравление датасетов. Исследовались получившие широкое практическое применение модели YOLOv5, YOLOv8, а также одна из новых моделей YOLOv11.

В отличие от ранее проведенных исследований, которые в основном направлены на оценку защищенности какой-либо одной из конкретных версий YOLO-моделей, в данном исследовании оценено семейство YOLO-детекторов. Показано, что в ходе эволюции моделей вплоть по текущих версий YOLOv11 улучшения моделей в основном направлены на добавление компонент, которые повышают точность обнаружения за счет многомасштабного обучения, однако YOLO-детекторы, по-прежнему, остаются подвержены глубоким градиентным состязательным атакам, таким как PGD-атаки.

Встроенные механизмы шумоподавления в YOLOv11 способны противостоять шумовым аддитивным атакам, а также градиентным атакам с малыми возмущениями, такими как FGSM, незначительно снижая производительность.

Предложенная в данной работе тактика комплексных состязательных атак, при которой состязательные примеры создаются с учетом взвешенных термов локализации, классификации и уверенности функции потерь в ходе трансферного обучения, позволила выявить, что наиболее эффективными является атаки на терм уверенности (деградация mAPval 50-95 от 20 при малом бюджете атаки до 100% при больших бюджетах для PGD-атак). Это обосновывает необходимость обязательного состязательного обучения, как меры защиты (Adversarial Defense), но подчеркивается недостаточность таких мер.

Литература

1. Ultralytics YOLO Docs, URL: https://docs.ultralytics.com/ru/models/, дата обращения 31.05.2025.

2. *Cao Y., Li C., Peng Y., Ru H.* MCS-YOLO: A Multiscale object detection method for autonomous driving road environment recognition // IEEE ACCESS, 2023. T. 11, C. 22342-22354. doi: 10.1109/ACCESS.2023.3252021

3. *Ren H., Jing F., Li S.* DCW-YOLO: Road object detection algorithms for autonomous driving // IEEE Access. 2024. T. 99. C. 1-1. doi: 10.1109/ACCESS.2024.3364681.

4. *Azevedo, P., Santos, V.* YOLO-based object detection and tracking for autonomous vehicles using edge devices. In: Tardioli D., Matellán V., Heredia G., Silva M.F., Marques, L. (eds) ROBOT2022: Fifth Iberian Robotics Conference // ROBOT 2022. Lecture Notes in Networks and Systems, 2023. T. 589. C. 297-308, Springer, Cham, doi: https://doi.org/10.1007/978-3-031-21065-5_25.

5. *Araújo B., Teixeira J., Fonseca J., Cerqueira R., Beco S.* The road to safety: A review of uncertainty and applications to autonomous driving perception // Entropy. 2024. T. 26. C. 634, doi: 10.3390/e26080634.

6. *Tu J., Ren M., Manivasagam S., Liang M., Yang B., Du R., Cheng F., Urtasun R.* (2020). Physically realizable adversarial examples for LiDAR object detection // IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), C. 13713-13722. doi: DOI:10.1109/CVPR42600.2020.01373

7. *Yang K., Tsai T., Yu H., Ho Ts-Yi., Jin Y.* Beyond digital domain: Fooling deep learning based recognition system in physical world // Proceedings of the AAAI Conference on Artificial Intelligence. 2020. T. 34. №. 1, C. 1088-1095. doi:10.1609/aaai.v34i01.5459

8. *Wang J., Liu A., Yin Z., Liu Sh., Tang Sh, Liu X.* Dual attention suppression attack: Generate adversarial camouflage in physical world // IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021. C. 8561-8570, doi: 10.1109/CVPR46437.2021.00846..

9. Vulnerabilities. CVE-2021-31681 – Deserialization of untrusted data vulnerability in ultralytics Yolov3. URL: https://cyber.vumetric.com/vulns/CVE-2021-31681/deserialization-of-untrusted-data-vulnerability-in-ultralytics-yolov3/, дата обращения 31.05.2025.

10. *Li R., Dai Sh-Ch., Xiong A.* WIP: Adversarial object-evasion attack detection in autonomous driving contexts: A simulation-based investigation using YOLO // Conference: Symposium on Vehicle Security & Privacy. 2024. doi:10.14722/vehiclesec.2024.23059.

11. *Ibrahum A.D.M., Hussain M., Hong JE.* Deep learning adversarial attacks and defenses in autonomous vehicles: a systematic literature review from a safety perspective // Artif Intell Rev 58. 2025. T. 28. URL: https://doi.org/10.1007/s10462-024-11014-8

12. *Sun Q., Yao X., Rao A., Yu B., Hu Sh.* Counteracting adversarial attacks in autonomous driving. // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2022 C. 1-1. doi: 10.1109/TCAD.2022.3166112.

13. *Фомичева С.Г., Беззатеев С.В.* Механизмы зашиты моделей машинного обучения от состязательных атак // T-Comm: Телекоммуникации и транспорт. 2023. Т. 17. № 10. С. 28-42. doi: 10.36724/2072-8735-2023-17-10-28-42.

14. *Jedh M., Othmane L.b., Somani A.K*. Improvement and evaluation of resilience of adaptive cruise control against spoofing attacks using intrusion detection system. In: Ait Wakrime A., Navarro-Arribas G., Cuppens F., Cuppens N., Benaini R. (eds) // Risks and Security of Internet and Systems. CRiSIS 2023. Lecture Notes in Computer Science, 2024. T. 14529. Springer, Cham. https://doi.org/10.1007/978-3-031-61231-2_5.

15. *Zhou X., Chen A., Kouzel M., Ren H. et al*. Runtime stealthy perception attacks against DNN-based adaptive cruise control systems. 2024. URL: https://www.researchgate.net/publication/377874677_Runtime_Stealthy _Perception_Attacks_against_DNN-based_Adaptive_Cruise_Control_Systems, дата обращения 31.05.2025.

16. *Беззатеев С.В., Фомичева С.Г., Жемелев Г.А*. Доверенное автоматическое машинное обучение при функционировании цифровых двойников // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18. № 7. С. 44-55, doi: 10.36724/2072-8735-2024-18-7-44-55.

17. *Bezzateev S.V., Zhemelev G.A., Fomicheva S.G.* Researching the performance of AutoML platforms in confidential computing // Automatic Control and Computer Sciences. 2024. Т. 58. № 8. С.1373-1385. doi: 10.3103/S0146411624701049

18. *Weilin Xu, et al.* Feature squeezing: Detecting adversarial examples in deep neural networks. URL: https://deepai.org/publication/feature-squeezing-detecting-adversarial-examples-in-deep-neural-networks, дата обращения 31.05.2025.

19. *Xie C., Wang J., Zhang Zh., Ren Zh., Yuille A*. Mitigating adversarial effects through randomization. // ArXiv abs/1711.01991 .2017: n. pag. doi: 10.48550/arXiv.1711.01991.

20. *Chen. T., Kornblith S., Norouzi M., Hinton G*. A Simple framework for contrastive learning of visual representations. 2020. doi: 10.48550/arXiv.2002.05709v3.

RESISTANCE RESEARCHING OF YOLO OBJECT DETECTORS TO ADVERSARIAL ATTACKS

Svetlana G. Fomicheva, *St. Petersburg University of Aerospace Instrumentations, St-Petersburg, Russia,* ***levikha@mail.ru***

Abstract

In physical world, object detectors encounter various interferences and noises, as well as malicious modifications that can significantly affect their performance. Such changes can lead to the removal of compromised detectors from their normal operating modes, which actualizes the tasks of monitoring existing and newly created YOLO models for resistance to adversarial attacks. The aim of the study is to assess the resilience of intelligent object detectors to complex adversarial attacks during transfer training. Machine learning and adversarial defense methods were used. The results of evaluating the performance of various YOLO family models against adversarial attacks aimed at gradients have shown that attacks on gradients of the confidence loss function are the most effective, leading to degradation of model performance from 20% with a small attack budget to 100% with large budgets of deep attacks on gradients. The tactics of a comprehensive attack on model gradients proposed in the paper substantiate the expediency of using competitive learning mechanisms as a mandatory protection measure, as well as additional measures such as training and operating verified YOLO models in trusted execution environments. It is not possible to counteract deep adversarial attacks on gradients with any one method of defence, the necessary and sufficient set of tools is determined by the specifics of the model operation.

Keywords: *object detector, loss function, adversarial attack, attack budget, performance of YOLO detectors*

References

[1] Ultralytics YOLO Docs. URL: https://docs.ultralytics.com/ru/models/,Date of access: 31.05.2025.

[2] Y. Cao., C. Li, Y. Peng, H. Ru, "MCS-YOLO: A Multiscale object detection method for autonomous driving road environment recognition," *IEEE Access,* 2023. vol. 11, pp. 22342-22354. DOI: 10.1109/ACCESS.2023.3252021

[3] H. Ren, F. Jing, S. Li, "DCW-YOLO: Road object detection algorithms for autonomous driving," *IEEE Access*. 2024. Vol. 99. pp. 1-1. DOI: 10.1109/ACCESS.2024.3364681.

[4] P. Azevedo, V. Santos, "YOLO-based object detection and tracking for autonomous vehicles using edge devices," In: D. Tardioli, V. Matellin, G. Heredia, M.F. Silva, L. Marques,(eds) ROBOT2022: Fifth Iberian Robotics Conference . ROBOT 2022. Lecture Notes in Networks and Systems, 2023. vol. 589. pp. 297-308, Springer, Cham, DOI: https://doi.org/10.1007/978-3-031-21065-5_25.

[5] B. Ara?jo, J. Teixeira, J. Fonseca, R. Cerqueira, S. Beco, "The road to safety: A review of uncertainty and applications to autonomous driving perception," *Entropy*. 2024. vol. 26.pp. 634, DOI: 10.3390/e26080634.

[6] J. Tu, M. Ren, S. Manivasagam, M .Liang, B. Yang, R. Du, F Cheng., R. Urtasun, "Physically realizable adversarial examples for LiDAR object detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13713-13722. DOI: DOI:10.1109/CVPR42600.2020.01373

[7] K. Yang, T. Tsai, H. Yu, Ts-Yi. Ho, Y. Jin, "Beyond digital domain: Fooling deep learning based recognition system in physical world," *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020. vol. 34. №. 1, pp. 1088-1095. DOI:10.1609/aaai.v34i01.5459.

[8] J. Wang, A. Liu, Z. Yin, Sh. Liu, Sh. Tang, X. Liu, "Dual attention suppression attack: Generate adversarial camouflage in physical world," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Nashville, TN, USA, 2021, pp. 8561-8570, DOI: 10.1109/CVPR46437.2021.00846.

[9] Vulnerabilities. CVE-2021-31681 – Deserialization of Untrusted Data vulnerability in Ultralytics Yolov3. URL: https://cyber.vumetric.com/vulns/CVE-2021-31681/deserialization-of-untrusted-data-vulnerability-in-ultralytics-yolov3/, Date of access 31.05.2025.

[10] R. Li, Sh-Ch. Dai, A. Xiong, "WIP: Adversarial object-evasion attack detection in autonomous driving contexts: A simulation-based investigation using YOLO." Conference: Symposium on Vehicle Security & Privacy. 2024. DOI:10.14722/vehiclesec.2024.23059.

[11] A.D.M. Ibrahum, M. Hussain, JE. Hong, "Deep learning adversarial attacks and defenses in autonomous vehicles: a systematic literature review from a safety perspective," *Artif Intell Rev 58*. 2025. vol. 28. URL: https://doi.org/10.1007/s10462-024-11014-8

[12] Q. Sun, X. Yao, A. Rao, B. Yu, Sh. Hu, "Counteracting adversarial attacks in autonomous driving," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2022 Pp. 1-1. DOI: 10.1109/TCAD.2022.3166112.

[13] S.G. Fomicheva, S.V. Bezzateev, "Defenses for machine learning models from adversarial attacks]," *T-Comm*. 2023. vl. 17. № 10. pp. 28-42. DOI: 10.36724/2072-8735-2023-17-10-28-42 (in Russian))

[14] M. Jedh, L.B. Othmane, A.K. Somani, "Improvement and evaluation of resilience of adaptive cruise control against spoofing attacks using intrusion detection system." In: A. Ait Wakrime, G. Navarro-Arribas, F. Cuppens, N. Cuppens, R. Benaini (eds). Risks and Security of Internet and Systems. CRiSIS 2023. Lecture Notes in Computer Science, 2024. Vol. 14529. Springer, Cham. https://doi.org/10.1007/978-3-031-61231-2_5.

[15] X. Zhou, A. Chen, M. Kouzel, H. Ren et al. "Runtime stealthy perception attacks against DNN-based adaptive cruise control systems," 2024. URL: https://www.researchgate.net/publication/377874677_Runtime_Stealthy_Perception _Attacks_against_DNN-based_Adaptive_Cruise_Control_ Systems, Date of access 31.05.2025.

[16] S.V Bezzateev., S.G. Fomicheva, G.A. Zhemelev, "Trusted automatic machine learning in the operation of digital twins," *T-Comm*, 2024. Vol. 18. No. 7. pp. 44-55, DOI: 10.36724/2072-8735-2024-18-7-44-55. (in Russian)

[17] S.V. Bezzateev, G.A. Zhemelev, S.G. Fomicheva, "Researching the performance of AutoML platforms in confidential computing," *Automatic Control and Computer Sciences*. 2024. vol. 58. № 8. pp. 1373-1385. DOI: 10.3103/S0146411624701049.

[18] Weilin Xu, et al. "Feature squeezing: Detecting adversarial examples in deep neural fetworks. "URL: https://deepai.org/publication/feature-squeezing-detecting-adversarial-examples-in-deep-neural-networks, Date of access 31.05.2025.

[19] C. Xie, J. Wang, Zh. Zhang, Zh Ren., A. Yuille, "Mitigating adversarial effects through randomization." ArXiv abs/1711.01991 .2017: n. pag. DOI: 10.48550/arXiv.1711.01991.

[20] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, "A Simple framework for contrastive learning of Visual representations," 2020. DOI: 10.48550/arXiv.2002.05709v3

Information about author:

Fomicheva S. G., *PhD, Full Professor, Professor at the Department of Information Security, St. Petersburg University of Aerospace Instrumentations, St. Petersburg, Russia*