

# Аннотация

Аугментация данных — важный инструмент современных исследователей в области детекции, позволяющий увеличить объём обучающей выборки. Однако существующие методы ограничены, поскольку не обеспечивают существенного семантического расширения данных. Это может привести к снижению способности моделей обобщать информацию. В этой работе предложен новый метод аугментации, основанный на семантической замене объектов на изображениях. Такой подход обеспечивает расширение обучающих выборок и повышает точность моделей детекции. Были проведены эксперименты, демонстрирующие влияние предложенного метода на функцию качества mAP, а также выполнен анализ влияния отдельных компонентов на данную функцию качества.

# Содержание

<b>1 Введение</b>	<b>4</b>
<b>2 Обзор литературы</b>	<b>6</b>
<b>3 Постановка задачи</b>	<b>8</b>
3.1 Функции потерь для задачи детекции . . . . .	9
3.2 Функции качества для задачи детекции . . . . .	11
3.3 Модель генеративной аугментации . . . . .	12
3.4 Ключевые утверждения . . . . .	13
<b>4 Методология</b>	<b>17</b>
4.1 Модель для детекции исходного объекта . . . . .	17
4.2 Модель для генерации текстового запроса . . . . .	18
4.2.1 Определение нового класса объекта . . . . .	18
4.2.2 Генерация описания сцены . . . . .	19
4.2.3 Формирование расширенного запроса . . . . .	19
4.3 Модель генерации нового объекта . . . . .	20
4.4 Модель фильтрации сгенерированного изображения . . . . .	21
<b>5 Вычислительные эксперименты</b>	<b>22</b>
<b>6 Заключение</b>	<b>25</b>

# 1 Введение

В современном машинном обучении одной из ключевых проблем является нехватка доступных данных. Объём и разнообразие выборки напрямую влияют на обобщающую способность моделей, однако сбор и разметка новых образцов требуют значительных временных и финансовых затрат. В таких условиях аугментация данных становится эффективным инструментом расширения тренировочного набора, позволяющим получать синтетические примеры, близкие к исходным.

Аугментация широко применяется во множестве прикладных задач машинного обучения. В области обработки естественного языка к классическим приёмам относятся переформулирование предложений, синонимичная замена слов или перестановка фраз. В задаче обработки аудио используются изменения скорости воспроизведения, сдвиг тона, добавление фонового шума и эха.

В компьютерном зрении распространены геометрические преобразования — повороты, отражения, масштабирование и обрезка, а также приёмы вроде добавления гауссовского шума или изменения яркости и контрастности изображения. В популярной сегментационной модели U-Net[18] для расширения датасета используют деформации с помощью случайных смещений. Тем не менее подобные методы не вносят семантических изменений и ограничены в разнообразии синтезируемых примеров: при поворотах и отражениях сохраняется исходная структура изображения, а при корректировке яркости и контраста меняется лишь цветовая палитра.

С появлением генеративных моделей на базе GAN(Generative Adversarial Networks)[7] и Stable Diffusion[17] научные исследования в области аугментации изображений получили новый импульс. GAN впервые позволили синтезиро-

вать реалистичные изображения, однако их обучение часто сопровождается нестабильностью, проблемами сходимости и артефактами. Диффузионные модели демонстрируют ведущие результаты в задаче синтеза высококачественных изображений (*state-of-the-art*), хотя и требуют больших вычислительных ресурсов. Применение этих методов для аугментации данных позволяет принципиально расширить семантическое разнообразие выборок за счет генерации новых объектов.

В этой работе мы предлагаем архитектуру для аугментации изображений в задаче детекции объектов. Наш вклад заключается в следующем:

1. Разработка автоматизированной архитектуры для аугментации.
2. Проведение экспериментов, демонстрирующих влияние аугментаций на целевую функцию качества модели детекции.
3. Анализ вклада отдельных компонентов архитектуры в итоговое качество.

Предложенный метод позволяет существенно разнообразить выборку, что значительно снижает затраты на ручной сбор данных и повышает устойчивость моделей к вариативности входных изображений. Таким образом, наш подход становится важным инструментом для исследователей в области детекции, которые стремятся быстро и эффективно получить данные без потери качества.

## 2 Обзор литературы

К числу наиболее ранних методов аугментации данных в компьютерном зрении относятся техники пространственных и цветовых преобразований, такие как повороты, смещения, изменение яркости и контрастности. В частности, в статье *Albumentations* [2] подробно описаны методы и их реализация для различных задач аугментации. Аналогичные подходы активно применялись в работах, посвящённых классическим моделям, таким как архитектуры, обученные на ImageNet [19], и U-Net [18], которые стали основой для многих современных систем в области компьютерного зрения.

С появлением генеративных моделей, таких как Stable Diffusion [17] и GAN [7], исследования в области аугментации данных для задач детекции вышли на качественно новый уровень.

Метод, предложенный в статье [11], использует диффузионные модели для генерации реалистичных фонов, что улучшает детекцию объектов. В работе [6] представлен подход для генерации объектов заданного класса с сохранением семантической согласованности. Метод в статье [8] также обеспечивает сохранение меток при генерации новых данных. Модель DiffusionEngine [24] объединяет генерацию изображений с автоматическим созданием разметки, что упрощает подготовку данных для детекции. Исследование AeroGen [21] демонстрирует эффективность диффузионных моделей для аугментации данных в задачах анализа спутниковых изображений. Подход Erase, then Redraw [14] предлагает инновационный метод замены фрагментов изображения с использованием диффузионных моделей. Обзорная статья [1] систематизирует современные методы аугментации на основе диффузионных моделей.

Альтернативой семантически направленной аугментации является подход Open Vocabulary Detection [25], в котором модель обучается обнаруживать

объекты с использованием языковых моделей. Однако этот метод ограничен предопределённым словарём и не позволяет динамически расширять разнообразие данных через уточнение свойств объектов.

Также стоит отметить архитектуру Garage для генеративной аугментации [4]. Я вместе с командой участвовал в разработке этой модели. Однако она не является автоматизированной и использует устаревшие модели.

### 3 Постановка задачи

Рассмотрим датасет для задачи детекции:

$$\mathcal{D} = \{(x_i, t_i), i = 1, \dots, n\},$$

где  $X$  — пространство изображений,  $x_i \in X$  — исходное изображение,  $T$  — пространство аннотаций отдельных объектов, каждая из которых содержит координаты ограничивающего прямоугольника и метку класса,  $\mathcal{F}(T) \subseteq 2^T$  — пространство аннотаций изображений множества  $X$ ,  $t_i \in \mathcal{F}(T)$  — множество аннотаций, соответствующих объектам на изображении  $x_i$ .

Рассмотрим произвольную модель детекции как отображение:

$$D : X \rightarrow \mathcal{F}(\hat{T}),$$

где  $\hat{T}$  — пространство аннотаций для отдельных объектов, содержащих координаты ограничивающих прямоугольников, классы объектов и уверенность, предсказанных моделью.  $\mathcal{F}(\hat{T}) \subseteq 2^{\hat{T}}$  — пространство предсказанных аннотаций изображений множества  $X$ .

Рассмотрим функцию IoU (Intersection over Union):

$$\text{IoU} : \hat{T} \times T \rightarrow [0,1],$$

которая рассчитывается по формуле:

$$\text{IoU}(\hat{a}, a) = \frac{|\hat{b} \cap b|}{|\hat{b} \cup b|},$$

где  $\hat{a}$  — предсказанная аннотация для одного объекта, содержащая координаты ограничивающего прямоугольника, класс и уверенность,  $a$  — истинная аннотация для одного объекта, содержащая координаты ограничивающего прямоугольника и класс,  $\hat{b} \in \hat{a}$  — предсказанный ограничивающий прямоугольник,  $b \in a$  — истинный ограничивающий прямоугольник.

Рассмотрим функцию GIoU (Generalized Intersection over Union):

$$\text{GIoU} : \hat{T} \times T \rightarrow [-1, 1],$$

которая рассчитывается по формуле:

$$\text{GIoU}(\hat{a}, a) = \text{IoU}(\hat{a}, a) - \frac{|\tilde{b} \setminus (\hat{b} \cup b)|}{|\tilde{b}|},$$

где  $\tilde{b}$  — минимальный по площади ограничивающий прямоугольник, содержащий оба  $\hat{b}$  и  $b$ .

### 3.1 Функции потерь для задачи детекции

Определим функцию потерь для модели детекции YOLO[16]  $f_\theta$ :

$$\begin{aligned} \mathcal{L}_{YOLO}(\theta) = & \lambda_{\text{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^K \mathbf{I}_{ij}^{\text{obj}} [(x_i^{gt} - \hat{x}_i)^2 + (y_i^{gt} - \hat{y}_i)^2] \\ & + \lambda_{\text{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^K \mathbf{I}_{ij}^{\text{obj}} \left[ (\sqrt{w_i^{gt}} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i^{gt}} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=1}^{S^2} \sum_{j=1}^K \mathbf{I}_{ij}^{\text{obj}} (\hat{C}_i - C_i)^2 + \lambda_{\text{noobj}} \sum_{i=1}^{S^2} \sum_{j=1}^K \mathbf{I}_{ij}^{\text{noobj}} (\hat{C}_i - C_i)^2 \\ & + \sum_{i=1}^{S^2} \mathbf{I}_i^{\text{obj}} \sum_{c \in \mathcal{C}} (\hat{p}_i(c) - p_i(c))^2, \end{aligned}$$

где  $S \times S$  — размер сетки, на которую разбивается изображение,  $K$  — количество предсказанных ограничивающих прямоугольников в каждой ячейке сетки,  $\lambda_{\text{coord}}, \lambda_{\text{noobj}}$  — коэффициенты, регулирующие вклад в функцию потерь,  $\mathbf{I}_{ij}^{\text{obj}}$  — индикатор наличия объекта в  $j$ -ом прямоугольнике  $i$ -й ячейки,  $\mathbf{I}_{ij}^{\text{noobj}}$  — индикатор отсутствия объекта в  $j$ -ом прямоугольнике  $i$ -й ячейки,  $(x_i^{gt}, y_i^{gt}, w_i^{gt}, h_i^{gt})$  — координаты центра, ширина и высота истинного

ограничивающего прямоугольника для  $i$ -й ячейки,  $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$  — предсказанные координаты ограничивающего прямоугольника для  $i$ -й ячейки,  $C_i$  и  $\hat{C}_i$  — истинная и предсказанная вероятность наличия объекта в  $i$ -й ячейке,  $\mathcal{C}$  — множество классов объектов,  $p_i(c)$  и  $\hat{p}_i(c)$  — истинная и предсказанная вероятность принадлежности объекта классу  $c$  для  $i$ -й ячейки.

YOLO представляет собой одностадийный детектор объектов, в котором единый проход по сети обеспечивает одновременное предсказание координат ограничивающих рамок и вероятностей классов для всего изображения. В основе архитектуры лежит глубокая свёрточная сеть, дополненная блоком обнаружения, формирующим выходные данные модели. Ключевым преимуществом YOLO является высокая вычислительная эффективность.

Определим функцию потерь для модели детекции DETR[3]  $g_\phi$ . В данной функции потерь реализуется алгоритм назначений для установления соответствия между предсказанными и истинными аннотациями:

$$\hat{\sigma} = \arg \min_{\sigma \in S_N} \sum_{i=1}^N \left[ -\mathbf{I}_{\{c_i \neq \emptyset\}} \hat{p}_{\sigma(i)}(c_i) + \mathbf{I}_{\{c_i \neq \emptyset\}} \left( \lambda_{L1} \|b_i - \hat{b}_{\sigma(i)}\|_1 + \lambda_{giou} \text{GIoU}(a_i, \hat{a}_{\sigma(i)}) \right) \right],$$

где  $\hat{\sigma}$  — оптимальное соответствие между истинными аннотациями и предсказанными,  $S_N$  — множество инъективных отображений из  $\{1, \dots, M\}$  в  $\{1, \dots, N\}$ ,  $M$  — число истинных аннотаций объектов на изображении,  $N > M$  — число предсказанных аннотаций объектов на изображении,  $\mathbf{I}_{\{c_i \neq \emptyset\}}$  — индикатор наличия объекта в истинном наборе,  $c_i$  — истинная метка класса объекта  $i$ ,  $\hat{p}_j(c)$  — предсказанная моделью вероятность класса  $c$  для аннотации  $j$ ,  $a_i$  — истинная аннотация объекта  $i$ ,  $b_i$  — истинный ограничивающий прямоугольник объекта  $i$ ,  $\hat{a}_j$  — предсказанная аннотация объекта  $j$ ,  $\hat{b}_j$  — предсказанный

ограничивающий прямоугольник объекта  $j$ ,  $\lambda_{L_1}$  и  $\lambda_{\text{giou}}$  — регуляризационные коэффициенты для задачи поиска оптимального соответствия.

После нахождения оптимального соответствия  $\hat{\sigma}$  мы можем ввести функцию потерь:

$$\begin{aligned} \mathcal{L}_{DETR}(\phi) = \sum_{i=1}^N & \left[ -\log \hat{p}_{\hat{\sigma}_\phi(i)}(c_i) + \mathbf{I}_{\{c_i \neq \emptyset\}} \left( \lambda_{L1} \|b_i - \hat{b}_{\hat{\sigma}_\phi(i)}\|_1 \right. \right. \\ & \left. \left. + \lambda_{\text{giou}} \text{GIoU}(a_i, \hat{a}_{\hat{\sigma}_\phi(i)}) \right) \right], \end{aligned}$$

DETR представляет собой одностадийный детектор объектов, объединяющий свёрточную сеть и энкодер–декодер на основе трансформера, в котором механизмы самовнимания обеспечивают взаимодействие между всеми частями изображения. В процессе детекции декодер–трансформер генерирует предсказания координат ограничивающих рамок и распределение вероятностей классов. Ключевым преимуществом DETR является эффективный учёт глобального контекста сцены, однако из-за тяжеловесной архитектуры трансформера скорость вывода модели остаётся относительно низкой.

Решаются следующие оптимизационные задачи:

$$\theta^* = \arg \min_{\theta} \mathcal{L}_{YOLO}(\theta), \quad \phi^* = \arg \min_{\phi} \mathcal{L}_{DETR}(\phi).$$

### 3.2 Функции качества для задачи детекции

Определим функции качества для задачи детекции. Рассмотрим функцию mAP (mean Average Precision).

$$\text{mAP} : \{\hat{T}\} \times \{T\} \times [0,1] \rightarrow [0,1],$$

Для каждого класса  $c \in \mathcal{C}$  вычисляется функция AP (Average Precision):

$$\text{AP}(c, \tau, t, \hat{t}) = \int_0^1 P_c(r, \tau, t, \hat{t}) dr,$$

где  $P_c(r, \tau, t, \hat{t})$  — функция, задающая кривую Precision–Recall для класса  $c$  при пороге  $\tau$ ,  $t \subseteq T$  — множество истинных разметок для класса  $c$ ,  $\hat{t} \subseteq \hat{T}$  — множество предсказанных разметок для класса  $c$ .

$$\text{mAP} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{AP}(c, \tau, t, \hat{t}).$$

Рассмотрим функцию  $\text{mAP}_{50:95}$ :

$$\text{mAP}_{50:95} : \{\hat{T}\} \times \{T\} \rightarrow [0,1],$$

Определим промежуточную функцию  $\text{AP}_{50:95}$  для каждого класса  $c$  как усреднение  $\text{AP}(c, \tau, t, \hat{t})$  по десяти порогам:

$$\text{AP}_{50:95}(c, t, \hat{t}) = \frac{1}{10} \sum_{\tau \in \{0.50, 0.55, \dots, 0.95\}} \text{AP}(c, \tau, t, \hat{t}).$$

Функция  $\text{mAP}_{50:95}$  усредняет  $\text{AP}_{50:95}(c, t, \hat{t})$  по всем классам:

$$\text{mAP}_{50:95}(t, \hat{t}) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{AP}_{50:95}(c, t, \hat{t}).$$

### 3.3 Модель генеративной аугментации

Рассмотрим модель генеративной аугментации как отображение:

$$F_{\psi, \alpha, \beta, \gamma} : X \times [0,1] \longrightarrow (X_{\text{aug}} \times T_{\text{aug}}) \cup \{\emptyset\},$$

$$f_\psi : X \rightarrow M \times L \times T_{\text{aug}}$$

$$g_\alpha : X \times L \rightarrow P$$

$$h_\beta : X \times M \times P \rightarrow X_{\text{aug}}$$

$$r_\gamma : Y \times M \times L \times [0,1] \rightarrow \{0,1\}$$

где  $X$  — пространство исходных изображений,  $X_{\text{aug}}$  — пространство аугментированных изображений,  $T_{\text{aug}}$  — пространство разметок аугментированных объектов на изображениях, отображение  $f_\psi$  — модель детекции объекта, который будет аугментирован, отображение  $g_\alpha$  — модель генерации текстового запроса для аугментации нового объекта, отображение  $h_\beta$  — модель генерации нового объекта, отображение  $r_\gamma$  — модель фильтрации некачественных генераций, где  $M$  — пространство бинарных масок объектов исходных изображений,  $P$  — пространство текстовых запросов для аугментации объекта,  $L \subset P$  — пространство классов объектов изображений, Число из отрезка  $[0,1]$  отвечает за порог для модели фильтрации.

$$F_{\psi,\alpha,\beta,\gamma}(x, \tau) = \begin{cases} (x_{\text{aug}}, a_{\text{aug}}), & \text{если } r_\gamma(x_{\text{aug}}, m, \ell, \tau) = 1, \\ \emptyset, & \text{если } r_\gamma(x_{\text{aug}}, m, \ell, \tau) = 0. \end{cases}$$

где  $(m, \ell, a_{\text{aug}}) = f_\psi(x)$ ,  $x_{\text{aug}} = h_\beta(x, m, g_\alpha(x, \ell))$ .

### 3.4 Ключевые утверждения

Пусть  $\mathcal{D} = \mathcal{D}_{\text{val}} \sqcup \mathcal{D}_{\text{train}}$ . Рассмотрим аугментированный датасет для задачи детекции:

$$\mathcal{D}_{\text{aug}}(\tau) = \{(x_i^{\text{aug}}, t_i^{\text{aug}}), i = 1, \dots, m\},$$

где  $(x_i, t_i) \in \mathcal{D}_{\text{train}}$  — пара «изображение-разметка изображения» из обучающего датасета,  $(x_i^{\text{aug}}, a_i^{\text{aug}}) = F_{\psi,\alpha,\beta,\gamma}(x_i, \tau)$  — пара «аугментированное изображение-разметка аугментированного объекта»,  $a_i^* \in t_i$  — аннотация объекта с наибольшей площадью ограничивающего прямоугольника,  $t_i^{\text{aug}} = (t_i \setminus \{a_i^*\}) \cup \{a_i^{\text{aug}}\}$  — разметка аугментированного изображения,  $\tau \in [0,1]$  — пороговое значение для модели фильтрации.

**Утверждение 1.** Пусть  $\mathcal{D}_{val} = \{(x_i, t_i), i = 1, \dots, k\}$ . Существует такое значение  $\tau^* \in [0,1]$ , что модели детекции  $f_{\theta_1}$  и  $g_{\phi_1}$ , обученные на обединённом датасете  $\mathcal{D}_{aug}(\tau^*) \sqcup \mathcal{D}_{train}$ , достигают не меньшего значения по функциям mAP и mAP<sub>50:95</sub> на  $\mathcal{D}_{val}$ , чем модели  $f_{\theta_2}$  и  $g_{\phi_2}$ , обученные на  $\mathcal{D}_{train}$ . То есть:

$$\begin{aligned} \text{mAP}\left(\{f_{\theta_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right) &\geq \text{mAP}\left(\{f_{\theta_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right), \\ \text{mAP}_{50:95}\left(\{f_{\theta_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right) &\geq \text{mAP}_{50:95}\left(\{f_{\theta_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right), \\ \text{mAP}\left(\{g_{\phi_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right) &\geq \text{mAP}\left(\{g_{\phi_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right), \\ \text{mAP}_{50:95}\left(\{g_{\phi_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right) &\geq \text{mAP}_{50:95}\left(\{g_{\phi_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right). \end{aligned}$$

Рассмотрим модель генеративной аугментации следующего вида:

$$F'_{\psi, \beta, \gamma}(x, \tau) = \begin{cases} (x_{\text{aug}}, a_{\text{aug}}), & \text{если } r_{\gamma}(x_{\text{aug}}, m, \ell, \tau) = 1, \\ \emptyset, & \text{если } r_{\gamma}(x_{\text{aug}}, m, \ell, \tau) = 0. \end{cases}$$

где  $(m, \ell, a_{\text{aug}}) = f_{\psi}(x)$ ,  $x_{\text{aug}} = h_{\beta}(x, m, \ell)$ .

Рассмотрим аугментированный датасет для задачи детекции:

$$\mathcal{D}'_{\text{aug}}(\tau) = \{(x_i^{\text{aug}}, t_i^{\text{aug}}), i = 1, \dots, n\},$$

где  $(x_i, t_i) \in \mathcal{D}_{train}$  — пара «изображение-разметка изображения» из обучающего датасета,  $(x_i^{\text{aug}}, a_i^{\text{aug}}) = F'_{\psi, \beta, \gamma}(x_i, \tau)$  — пара «аугментированное изображение-разметка аугментированного объекта»,  $a_i^* \in t_i$  — аннотация объекта с наибольшей площадью ограничивающего прямоугольника,  $t_i^{\text{aug}} = (t_i \setminus \{a_i^*\}) \cup \{a_i^{\text{aug}}\}$  — разметка аугментированного изображения,  $\tau \in [0,1]$  — пороговое значение для модели фильтрации.

**Утверждение 2.** Пусть  $\mathcal{D}_{val} = \{(x_i, t_i), i = 1, \dots, k\}$ . Существует такое значение  $\tau^* \in [0,1]$ , что модели детекции  $f_{\theta_1}$  и  $g_{\phi_1}$ , обученные на обединённом датасете  $\mathcal{D}_{aug}(\tau^*) \sqcup \mathcal{D}_{train}$ , достигают не меньшего значения по функциям

$\text{mAP}$  и  $\text{mAP}_{50:95}$  на  $\mathcal{D}_{val}$ , чем модели  $f_{\theta_2}$  и  $g_{\phi_2}$ , обученные на  $\mathcal{D}'_{aug}(\tau^*) \sqcup \mathcal{D}_{train}$ .  
To есть:

$$\begin{aligned}\text{mAP}\left(\{f_{\theta_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right) &\geq \text{mAP}\left(\{f_{\theta_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right), \\ \text{mAP}_{50:95}\left(\{f_{\theta_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right) &\geq \text{mAP}_{50:95}\left(\{f_{\theta_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right), \\ \text{mAP}\left(\{g_{\phi_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right) &\geq \text{mAP}\left(\{g_{\phi_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right), \\ \text{mAP}_{50:95}\left(\{g_{\phi_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right) &\geq \text{mAP}_{50:95}\left(\{g_{\phi_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right).\end{aligned}$$

Аналогично рассмотрим модель генеративной аугментации следующего вида:

$$F''_{\psi, \alpha, \beta}(x, \tau) = (x_{\text{aug}}, a_{\text{aug}})$$

где  $(m, \ell, a_{\text{aug}}) = f_{\psi}(x)$ ,  $x_{\text{aug}} = h_{\beta}(x, m, g_{\alpha}(x, \ell))$ .

Рассмотрим аугментированный датасет для задачи детекции:

$$\mathcal{D}''_{\text{aug}}(\tau) = \{(x_i^{\text{aug}}, t_i^{\text{aug}}), i = 1, \dots, n\},$$

где  $(x_i, t_i) \in \mathcal{D}_{train}$  — пара «изображение-разметка изображения» из обучающего датасета,  $(x_i^{\text{aug}}, a_i^{\text{aug}}) = F''_{\psi, \alpha, \beta}(x_i, \tau)$  — пара «аугментированное изображение-разметка аугментированного объекта»,  $a_i^* \in t_i$  — аннотация объекта с наибольшей площадью ограничивающего прямоугольника,  $t_i^{\text{aug}} = (t_i \setminus \{a_i^*\}) \cup \{t_{\text{aug}}\}$  — разметка аугментированного изображения,  $\tau \in [0, 1]$  — пороговое значение для модели фильтрации.

**Утверждение 3.** Пусть  $\mathcal{D}_{val} = \{(x_i, t_i), i = 1, \dots, k\}$ . Существует такое значение  $\tau^* \in [0, 1]$ , что модели детекции  $f_{\theta_1}$  и  $g_{\phi_1}$ , обученные на обединённом датасете  $\mathcal{D}_{\text{aug}}(\tau^*) \sqcup \mathcal{D}_{train}$ , достигают не меньшего значения по функциям  $\text{mAP}$  и  $\text{mAP}_{50:95}$  на  $\mathcal{D}_{val}$ , чем модели  $f_{\theta_2}$  и  $g_{\phi_2}$ , обученные на  $\mathcal{D}''_{\text{aug}}(\tau^*) \sqcup \mathcal{D}_{train}$ .  
To есть:

$$\begin{aligned}
& \text{mAP}\left(\{f_{\theta_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right) \geq \text{mAP}\left(\{f_{\theta_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right), \\
& \text{mAP}_{50:95}\left(\{f_{\theta_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right) \geq \text{mAP}_{50:95}\left(\{f_{\theta_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right), \\
& \text{mAP}\left(\{g_{\phi_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right) \geq \text{mAP}\left(\{g_{\phi_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k, 0.5\right), \\
& \text{mAP}_{50:95}\left(\{g_{\phi_1}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right) \geq \text{mAP}_{50:95}\left(\{g_{\phi_2}(x_i)\}_{i=1}^k, \{t_i\}_{i=1}^k\right).
\end{aligned}$$

## 4 Методология

Предложенная модель генеративной аугментации включает несколько компонентов: модель детекции, модель описания нового объекта, модель генерации нового объекта и модель фильтрации.

### 4.1 Модель для детекции исходного объекта

Наша архитектура реализована в полностью автоматическом режиме. Для первичной детекции объекта применяется предварительно обученная модель YOLO, способная обнаруживать объекты и строить для каждого найденного объекта ограничивающий прямоугольник с указанием класса. Среди всех возвращённых ограничивающих прямоугольников система выбирает один с максимальной площадью, что гарантирует фокусировку на наиболее значимом объекте сцены для последующей аугментации.

Результатом работы детекции являются координаты bbox выбранного для аугментации объекта и его класс, необходимые для передачи в последующие модули архитектуры.

Если пользователь желает аугментировать конкретный объект, он может передать системе маску в формате одноканального изображения, указывающую область для аугментации. В этом случае YOLO не выполняет автоматический выбор и использует переданный ограничивающий прямоугольник. Также пользователю потребуется передать еще класс объекта, которому соответствует переданная маска.

## 4.2 Модель для генерации текстового запроса

Процесс генерации текстового запроса разбит на несколько последовательных этапов, обеспечивающих корректное семантическое соответствие исходного и нового объектов, а также учёт визуального контекста изображения.

### 4.2.1 Определение нового класса объекта

Сначала архитектура получает метку исходного класса объекта из выходных данных детектора YOLO. Затем применяется zero-shot классификатор [23] для выбора наиболее вероятного релевантного обобщающего класса (например, «air vehicle», «land animal» и т.д.), используя текстовый запрос, содержащий только метку исходного объекта.

Далее задача заключается в выборе нового класса из списка кандидатов. Для этого используется тот же zero-shot классификатор с запросом в формате ’’What {class\_type} is similar to a {source\_class}?’’ что позволяет модели выбирать правдоподобные замены из заданного списка. Чтобы получить независимые вероятности для каждой метки, включается режим multilabel, при котором вероятность каждой метки рассчитывается отдельно, без нормировки по сумме вероятностей всех кандидатов. В результате формируется вектор оценок—вероятностей для каждого кандидата на замену, после чего отбираются все метки с оценкой выше 0,4 в список. Если после этого он оказывается пустым, то выбирается метка с максимальной вероятностью, что гарантирует наличие замены. Из оставшихся кандидатов система случайным образом выбирает один новый класс, обеспечивая вариативность и использование семантически близких меток. Такой подход позволяет учитывать произвольное число кандидатов и захватывать смысловые пересечения с исходным объектом.

Стоит отметить, что пользователь может самостоятельно задавать список кандидатов для выбора, а также перечень обобщающих классов.

#### 4.2.2 Генерация описания сцены

Для того чтобы модель аугментации учитывала более широкий визуальный контекст и семантические детали, данный этап включает генерацию текстового описания всего изображения с учётом исходного класса объекта. В качестве инструмента используется BLIP[10], обученный на наборе пар «изображение–текст» и способный генерировать описания, отражающие содержимое сцены и особенности объектов. При подаче исходного класса BLIP формирует текстовый запрос, подробно описывающий сцену, что позволяет модели получить представление о визуальных признаках и взаимосвязях между объектами.

Стоит отметить, что при передаче сформированного описания в следующую компоненту архитектуры название исходного объекта намеренно скрывается — этот приём используется для того, чтобы модель сосредоточилась на генерации нового объекта и не опиралась на информацию о предыдущем.

#### 4.2.3 Формирование расширенного запроса

После выбора нового класса (`new_object`) и формирования описания изображения (`image_description`) модель Qwen3-8B[22] генерирует расширенный запрос для дальнейшей аугментации. Конкретный шаблон запроса выглядит следующим образом:

```
USER: Write a concise, realistic visual description of a {new_object}
in less than 20 words in that scene: {image_description}.
Don't include background or other objects. Use descriptive terms only
```

about the `{new_object}`.

Then append style comments like: "4k, ultra HD, highly detailed, realistic lighting".

ASSISTANT:

где `{new_object}` заменяется на семантически подобранный новый класс, а `{image_description}` — на ранее сгенерированное описание изображения. При передаче такого запроса Qwen3-8B выдаёт развёрнутую визуальную подсказку, содержащую лаконичное описание нового объекта в контексте сцены и рекомендации по стилю (качество изображения, детализация, освещение и т. п.), что обеспечивает модели генерации объектов необходимую семантическую и визуальную информацию для генерации реалистичных изображений.

### 4.3 Модель генерации нового объекта

Основным компонентом архитектуры является модель генерации нового объекта — FLUX[9]. Эта модель относится к семейству диффузионных и обучается с помощью функции потерь, основанной на Flow Matching[13].

Рассмотрим неизвестное распределение данных  $q(x)$  с доступным набором выборок из этого распределения. Введём непрерывный путь плотностей  $\{p_t\}_{t \in [0,1]}$ , где  $p_0(x) = \mathcal{N}(x | 0, I)$ , а  $p_1(x) \approx q(x)$ .

Flow Matching нацелено на то, чтобы обучить параметрическое векторное поле  $v_\omega(x, t)$  так, чтобы оно совпадало с истинным векторным полем  $u_t(x)$ , которое задаёт эволюцию плотностей  $p_t$ . Тогда функция потерь Flow Matching записывается следующим образом:

$$\mathcal{L}_{\text{FM}}(\omega) = \mathbb{E}_{\substack{t \sim \mathcal{U}(0,1) \\ x \sim p_t}} \| v_\omega(x, t) - u_t(x) \|_2^2$$

где  $\omega$  — параметры нейросети,  $t$  точка из равномерного распределения на  $[0,1]$ ,  $x$  — точка, сгенерированная из плотности  $p_t$ .

При обучении решается следующая оптимизационная задача:

$$\omega^* = \arg \min_{\omega} \mathcal{L}_{FM}(\omega)$$

На практике проблема в том, что  $p_t$  и  $u_t$  обычно неизвестны в явном виде. Для каждого примера из датасета  $x_0 \sim p_0$ ,  $x_1 \sim p_1$  можно строить локальный путь и локальное поле, а затем агрегировать их, чтобы получить приближённые  $p_t$  и  $u_t$ . Это даёт более удобный и вычислительно эффективный подход к Flow Matching. Подробнее можно ознакомиться в статье[13].

#### 4.4 Модель фильтрации сгенерированного изображения

После генерации аугментированных изображений применяется модель фильтрации AlphaCLIP[20], разработанная как модификация архитектуры CLIP[15]. В отличие от базовой модели, AlphaCLIP позволяет вычислять семантическое соответствие не только между полным изображением и текстом, но и между локальными областями изображения (заданными бинарной маской) и текстовым описанием объекта.

Для оценки качества сгенерированных аугментаций вычисляется скалярное произведение между латентным представлением текстового описания целевого объекта и латентным представлением области изображения, выделенной с помощью сегментационной маски.

Порог семантического сходства задается пользователем: его регулировка позволяет контролировать строгость фильтрации. Уменьшение порога понижает качество аугментаций.

## 5 Вычислительные эксперименты

В качестве основных датасетов для задачи детекции были выбраны Pascal VOC[5] и COCO[12]. Датасет Pascal VOC содержит 20 классов объектов, тогда как COCO включает 80 классов.

При генерации аугментированного датасета на основе Pascal VOC и COCO мы заменили объект с наибольшим ограничивающим прямоугольником новым объектом другого класса, выбранным из списка меток Pascal VOC и COCO соответственно. При этом класс исходного объекта заранее удалялся из списка кандидатов, чтобы он не мог быть выбран повторно. Пороговое значение для модели фильтрации составляло 0.23.

Для получения экспериментальных результатов были использованы модели YOLOv1n и RTDETR-L из библиотеки Ultralytics. Проведён анализ влияния аугментаций на функцию качества mAP и mAP50:95, а также исследовано влияние отдельных компонентов архитектуры на итоговые значения mAP и mAP50:95. Обучение каждой из моделей детекции выполнялось в течение 500 эпох с нуля, после чего сравнивались максимальные показатели mAP и mAP50:95 на валидационном датасете. Результаты экспериментов приведены в соответствующей таблице 1.

Ниже представлены примеры объектов из датасетов: на рисунке 1 — исходные изображения, на рисунке 2 — результаты их аугментации, а на рисунке 3 — полученные изображения без отдельных компонентов архитектуры.

Данные вычислительные эксперименты показывают, что аугментации влияют на разнообразие датасета и улучшают показатель mAP. Кроме того, продемонстрировано, что модель расширения текстового запроса и модель фильтрации являются важными компонентами нашей архитектуры и также влияют на итоговое значение mAP.



Рис. 1: На левом изображении представлен объект класса «собака»; на правом — объекты класса «корова» из оригинального датасета.

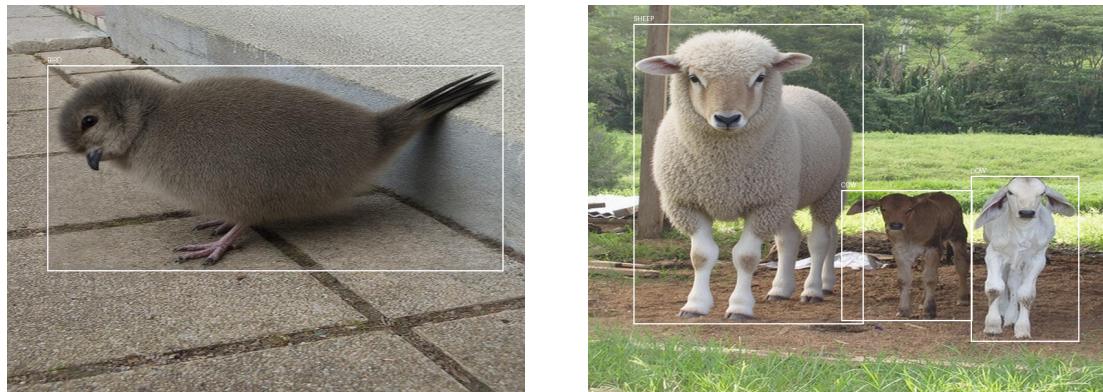


Рис. 2: На левом изображении представлен аугментированный объект класса «птица»; на правом — аугментированный объект класса «овца» и оригинальные объекты класса «корова».

Dataset	Model	Setting	Size	mAP@50	mAP@95
Pascal VOC	DETR	original	4000	57.2	41.2
		w/o expanded prompt	4000 + 4000	55.4	38.7
		w/o filter model	4000 + 4000	57.4	40.9
		ours	4000 + 4000	<b>58.2</b>	<b>41.4</b>
COCO	YOLO	original	4000	59.6	41.5
		w/o expanded prompt	4000 + 4000	59.4	41.2
		w/o filter model	4000 + 4000	61.4	<b>43.2</b>
		ours	4000 + 4000	<b>61.5</b>	<b>43.2</b>
	DETR	original	5000	26.6	17.6
		w/o expanded prompt	5000 + 5000	27.5	<b>17.8</b>
		w/o filter model	5000 + 5000	26	16.5
		ours	5000 + 5000	<b>27.8</b>	<b>17.8</b>
	YOLO	original	5000	26.7	17.4
		w/o expanded prompt	5000 + 5000	27.5	17.9
		w/o filter model	5000 + 5000	27.7	17.9
		ours	5000 + 5000	<b>28.2</b>	<b>18.3</b>

Таблица 1: Сравнение моделей DETR и YOLO, обученных на данных с аугментациями и без, а также анализ влияния отдельных компонентов на значение функций качества mAP.

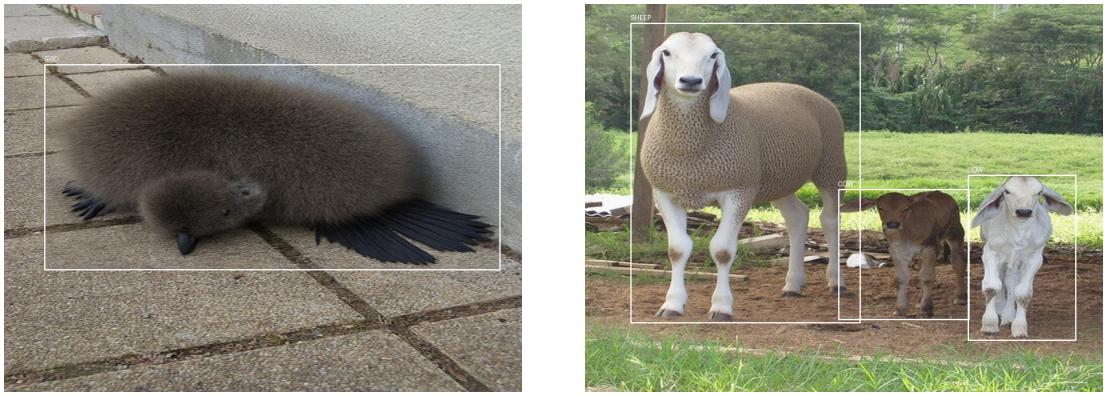


Рис. 3: На левом изображении представлен аугментированный объект класса «птица», созданный моделью без учёта компонента фильтрации; на правом — аугментированный объект класса «овца» вместе с оригинальными объектами класса «корова», сгенерированный без расширенной подсказки.

## 6 Заключение

В данной работе представлен полностью автоматизированный метод аугментации данных для задач детекции. Разработанный метод вносит весомый вклад в область аугментации данных, преодолевая ключевые недостатки существующих решений.

Одним из основных ограничений предлагаемого подхода является высокая вычислительная сложность: для генерации каждого аугментированного изображения требуются значительные ресурсы, включая мощные графические процессоры и продолжительное время выполнения. Подобная проблема характерна для методов, основанных на архитектуре диффузионных моделей.

Результаты экспериментов показывают, что модели, обученные на оригинальных и аугментированных данных, демонстрируют более высокие показатели качества. Кроме того, автоматизация процесса делает метод удобным и доступным средством подготовки данных.

## Список литературы

- [1] Alimisis, P., I. Mademlis, P. Radoglou-Grammatikis, P. Sarigiannidis, and G. T. Papadopoulos (2025). Advances in diffusion models for image data augmentation: A review of methods, models, evaluation metrics and future research directions.
- [2] Buslaev, A. V., A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin (2018). Albumentations: fast and flexible image augmentations. *CoRR abs/1809.06839*.
- [3] Carion, N., F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko (2020). End-to-end object detection with transformers. *CoRR abs/2005.12872*.
- [4] Daniil, D. and et.al (2024). Garage. <https://github.com/DorinDaniil/Garage>.
- [5] Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [6] Fang, H., B. Han, S. Zhang, S. Zhou, C. Hu, and W.-M. Ye (2024). Data augmentation for object detection via controllable diffusion models. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1246–1255.
- [7] Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial networks.
- [8] Kupyn, O. and C. Rupprecht (2024). Dataset enhancement with instance-level augmentations.

- [9] Labs, B. F. (2024). Flux. <https://github.com/black-forest-labs/flux>.
- [10] Li, J., D. Li, C. Xiong, and S. C. H. Hoi (2022). BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. *CoRR abs/2201.12086*.
- [11] Li, Y., X. Dong, C. Chen, W. Zhuang, and L. Lyu (2024). A simple background augmentation method for object detection with diffusion model.
- [12] Lin, T., M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft COCO: common objects in context. *CoRR abs/1405.0312*.
- [13] Lipman, Y., R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le (2023). Flow matching for generative modeling.
- [14] Ma, F., W. Qi, G. Zhao, M. Liu, and J. Ma (2025). Erase, then redraw: A novel data augmentation approach for free space detection using diffusion model.
- [15] Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever (2021). Learning transferable visual models from natural language supervision. *CoRR abs/2103.00020*.
- [16] Redmon, J., S. K. Divvala, R. B. Girshick, and A. Farhadi (2015). You only look once: Unified, real-time object detection. *CoRR abs/1506.02640*.
- [17] Rombach, R., A. Blattmann, D. Lorenz, P. Esser, and B. Ommer (2021). High-resolution image synthesis with latent diffusion models. *CoRR abs/2112.10752*.
- [18] Ronneberger, O., P. Fischer, and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR abs/1505.04597*.

- [19] Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei (2014). Imagenet large scale visual recognition challenge. *CoRR abs/1409.0575*.
- [20] Sun, Z., Y. Fang, T. Wu, P. Zhang, Y. Zang, S. Kong, Y. Xiong, D. Lin, and J. Wang (2023). Alpha-clip: A clip model focusing on wherever you want.
- [21] Tang, D., X. Cao, X. Wu, J. Li, J. Yao, X. Bai, D. Jiang, Y. Li, and D. Meng (2025). Aerogen: Enhancing remote sensing object detection with diffusion-driven data generation.
- [22] Yang, A., A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu (2025). Qwen3 technical report.
- [23] Yin, W., J. Hay, and D. Roth (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *CoRR abs/1909.00161*.
- [24] Zhang, M., J. Wu, Y. Ren, M. Li, J. Qin, X. Xiao, W. Liu, R. Wang, M. Zheng, and A. J. Ma (2023). Diffusionengine: Diffusion model is scalable data engine for object detection.
- [25] Zhu, C. and L. Chen (2024). A survey on open-vocabulary detection and segmentation: Past, present, and future.