

## **Аннотация**

Целью данной работы является разработка рекуррентной нейронной сети на основе архитектуры YOLO для детекции документов на последовательности кадров. Основные задачи включают разработку и обучение базовой модели YOLOv3, реализацию алгоритмического трекинга для улучшения детекции объектов в видео-потоках, а также интеграцию рекуррентного слоя в YOLOv3 для учета временных зависимостей между кадрами.

В ходе исследования была создана модель на основе YOLOv3, которая демонстрирует высокую точность детекции объектов на отдельных кадрах. Для улучшения анализа видеопоследовательностей был реализован алгоритмический трекинг. Интеграция рекуррентного слоя (GRU) в модель YOLOv3 позволила учитывать временные зависимости между кадрами, что улучшило качество детекции объектов в видеопотоках.

На основании данной работы рекомендуется использовать предложенную архитектуру для задач, требующих высокой точности и стабильности детекции объектов в реальном времени, таких как системы видеонаблюдения, автономноеождение и распознавание документов. Будущие исследования могут быть направлены на оптимизацию модели для различных типов данных и улучшение алгоритмов трекинга, а также на применение других рекуррентных слоев для дальнейшего повышения точности детекции.

# Содержание

<b>1 Обозначения и сокращения</b>	<b>4</b>
<b>2 Введение</b>	<b>4</b>
<b>3 Обзор исследования</b>	<b>5</b>
3.1 Обзор литературы . . . . .	5
3.2 Определение объекта и предмета исследования . . . . .	6
3.3 Цели и задачи исследования . . . . .	6
3.4 Датасет . . . . .	6
3.5 Функция качества . . . . .	8
3.6 Функция ошибки (Loss) . . . . .	8
3.7 Оптимизатор . . . . .	9
3.8 Обучение и валидация . . . . .	9
3.9 Методы исследования . . . . .	9
3.10 Научно-теоретическая и практическая значимость исследования . . . . .	9
<b>4 Изменение датасета</b>	<b>11</b>
<b>5 Базовая модель</b>	<b>12</b>
5.1 Результаты . . . . .	14
<b>6 Реализация алгоритмического трекинга</b>	<b>15</b>
6.1 Результаты . . . . .	15
<b>7 Разработка рекуррентной YOLO</b>	<b>16</b>
7.1 Изменение структуры датасета . . . . .	16
7.1.1 Преимущества изменений . . . . .	16
7.2 Аугментации данных . . . . .	17
7.3 Результаты . . . . .	19
<b>8 Эксперименты</b>	<b>20</b>
8.1 Неправильное разделение датасета . . . . .	20
8.1.1 С аугментациями . . . . .	20
8.2 Правильное разделение датасета . . . . .	21
8.2.1 Без аугментаций . . . . .	21
8.2.2 С аугментациями . . . . .	24
8.3 Общие результаты . . . . .	25
<b>9 Заключение</b>	<b>26</b>
<b>Список литературы</b>	<b>28</b>

# 1 Обозначения и сокращения

Обозначение	Расшифровка
YOLO	You Only Look Once (алгоритм детекции объектов)
IoU	Intersection over Union (метрика качества для оценки пересечения предсказанной и истинной рамок)
GRU	Gated Recurrent Unit (вариант рекуррентной нейронной сети)
BCE	Binary Cross-Entropy (бинарная кросс-энтропия, функция потерь)
MSE	Mean Squared Error (среднеквадратичная ошибка, функция потерь)
ReLU	Rectified Linear Unit (функция активации)
SymReLU	Symmetrical Rectified Linear Unit (симметричная функция активации ReLU)
PyTorch	Фреймворк для глубокого обучения
Adam	Оптимизатор Adam (адаптивный метод момента)

## 2 Введение

В условиях стремительного развития технологий видеоанализа и компьютерного зрения, задача детекции объектов на последовательности кадров становится особенно актуальной в таких областях, как системы видеонаблюдения, автономное вождение и распознавание документов. Использование рекуррентных нейронных сетей в сочетании с передовыми архитектурами свёрточных нейронных сетей, такими как YOLO, открывает новые возможности для точного и эффективного обнаружения объектов в реальном времени.

Цель исследования – разработка системы для эффективной детекции объектов на последовательности кадров с использованием архитектуры YOLO и интеграции рекуррентных слоев. Задачи исследования включают разработку и обучение базовой модели на основе YOLOv3, реализацию алгоритмического трекинга для улучшения детекции объектов в видеопотоках и интеграцию рекуррентного слоя для учета временных зависимостей между кадрами.

Методы исследования включают анализ научно-технической литературы, модификацию датасета, разработку и тестирование архитектур нейронных сетей с использованием PyTorch, проведение экспериментов с различными конфигурациями сетей и аугментациями данных, а также сравнительный анализ результатов моделей.

Научно-теоретическая значимость работы заключается в разработке новой архитектуры нейронной сети, объединяющей свёрточные и рекуррентные компоненты для повышения точности и эффективности детекции объектов на видеопоследовательностях. Практическая значимость состоит в создании системы, способной применяться в реальных условиях для задач видеонаблюдения и детекции документов, что значительно улучшает качество и надежность этих систем.

### 3 Обзор исследования

#### 3.1 Обзор литературы

Исследование "You Only Look Once: Unified, Real-Time Object Detection"(YOLO) [1], проведённое Redmon и соавторами, представляет значительный прорыв в области компьютерного зрения. YOLO переосмысливает задачу детекции объектов, представляя её как задачу регрессии для пространственно разделённых ограничивающих рамок и соответствующих вероятностей классов. Основные преимущества YOLO включают высокую скорость обработки, глобальный подход к анализу изображения и обобщаемость на различных типах изображений.

Исследование "Large-Scale Video Classification with Convolutional Neural Networks"[2], проведённое Karpathy и соавторами, фокусируется на применении свёрточных нейронных сетей для классификации видео в крупном масштабе. Основные моменты исследования включают масштабирование CNN для видео, архитектуры слияния временной информации и многоуровневые архитектуры.

В исследовании "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description"(LRCN) [3], проведённом Donahue и соавторами, фокусируется на интеграции свёрточных нейронных сетей (CNN) и рекуррентных нейронных сетей (RNN) для обработки и анализа видеоданных. LRCN объединяет CNN для извлечения визуальных признаков и рекуррентную сеть для моделирования временных зависимостей.

Исследование "Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution"[4], проведённое Qiao и соавторами, представляет новаторский подход к улучшению производительности моделей детекции объектов. Авторы вводят концепцию рекурсивной пирамиды признаков и переключаемой Atrous свёртки, что улучшает способность модели обнаруживать объекты разных масштабов и улучшает детекцию в условиях сложных и разнообразных сцен.

В исследовании "Анализ особенностей использования стационарных и мобильных малоразмерных цифровых видеокамер для распознавания документов"[5], проведённом Арлазаровым и соавторами, обсуждаются проблемы и возможности использования различных типов цифровых камер для распознавания текстовых документов. В исследовании сравниваются стационарные камеры и мобильные камеры по их возможностям и ограничениям для задач распознавания документов.

Исследование "Алгоритмы поиска границ печатных символов, используемые при оптическом распознавании символов"[6], выполненное Арлазаровым и соавторами, посвящено методам сегментации символов для оптического распознавания текста (OCR). В работе рассматриваются различные подходы к корректной сегментации символов, что является критическим этапом в процессе распознавания текста.

Книга "Multiple View Geometry in Computer Vision"[7], написанная Ричардом Хартли и Эндрю Зиссерманом, является фундаментальным трудом в области компьютерного зрения, посвящённым геометрии множественных видов. Она охватывает шир-

рокий спектр тем, связанных с анализом изображений, полученных с различных точек зрения, и предлагает математические методы для их обработки и интерпретации.

Исследование "Small Object Intelligent Detection Method Based on Adaptive Recursive Feature Pyramid"[8], проведённое Zhang и соавторами, предлагает новый метод для улучшения точности детекции малых объектов с использованием адаптивной рекурсивной сети пирамиды признаков (AR-PANet). Адаптивная рекурсивная структура и методы фьюзии признаков могут быть использованы для улучшения точности детекции объектов, особенно малых объектов.

Исследование "Recurrent Neural Networks for Video Object Detection"[9], проведённое Ahmad и Pettirsch, анализирует применение рекуррентных нейронных сетей (RNN) для детекции объектов в видео. Авторы сравнивают различные методы, включая те, которые используют карты признаков, уровни ограничивающих рамок и поточечные сети, что позволяет учитывать временной контекст и улучшать точность детекции объектов.

### **3.2 Определение объекта и предмета исследования**

**Объект исследования:** системы детекции объектов на последовательности кадров.

**Предмет исследования:** архитектуры нейронных сетей для детекции документов на видеопоследовательностях, включающие свёрточные и рекуррентные компоненты.

### **3.3 Цели и задачи исследования**

**Цель исследования:** разработка системы, способной эффективно детектировать объекты на последовательности кадров, используя архитектуру YOLO с интеграцией рекуррентных слоев.

**Задачи исследования:**

1. Разработка и обучение базовой модели детекции на основе YOLOv3.
2. Реализация алгоритмического трекинга для улучшения детекции объектов в видеопотоках.
3. Интеграция рекуррентного слоя в YOLOv3 для учета временных зависимостей между кадрами.

### **3.4 Датасет**

Для исследования был использован датасет MIDV-2020 [11]. Он включает в себя 1000 видеоклипов с документами удостоверяющими личность. Документы - это ID карты Албании, Испании, Эстонии, Финляндии, Словакии, и паспорта Азербайджана, Греции, Латвии, России и Сербии.

Съемка видеороликов производилась с помощью iPhone XR и Samsung S10 в 10 различных условиях:

1. Условия низкой освещенности;
2. На фоне клавиатуры;
3. Съемка на улице при естественном освещении;
4. На фоне стола;
5. На фоне тканей различных текстур;
6. На фоне текстового документа;
7. Сильные проективные искажения документа;
8. Блик от солнца или лампы скрывает часть документа.

Примеры каждого из условий представлены ниже.



Рисунок 3.1 – Примеры изображений из датасета MIDV-2020. Взято из источника [10]

Видео разбиты на кадры, к каждому видео есть аннотация. Аннотация содержит координаты ограничивающей прямоугольной границы документа, первая вершина четырехугольника соответствует верхнему левому углу физического документа, а остальные вершины идут по порядку по часовой стрелке. Также аннотация содержит другие данные, но они не нужны для данного исследования. Примеры изображений из датасета:



ID карта Албании



Паспорт России



ID карта Испании

Рисунок 3.2 – Примеры изображений из датасета MIDV-2020

### 3.5 Функция качества

Для оценки качества предсказания модели использовалась функция качества Intersection over Union (IoU). Эта метрика измеряет степень совпадения между предсказанными ограничивающими рамками и истинными рамками.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

где Area of Overlap — площадь пересечения предсказанной и истинной рамок, а Area of Union — площадь объединения этих рамок. Чем выше значение IoU, тем точнее предсказание модели.

### 3.6 Функция ошибки (Loss)

В качестве функции потерь использовалась комбинированная функция, включающая среднеквадратичную ошибку для координат ограничивающих рамок и бинарную кросс-энтропию для уверенности предсказания.

$$\begin{aligned} \text{Loss} = & \text{BCEWithLogitsLoss}(\text{pred\_confidences}, \text{target\_confidences}) \\ & + \text{MSELoss}(\text{pred\_boxes}, \text{target\_boxes}) \end{aligned} \quad (1)$$

где pred\\_confidences — предсказанные уверенности, target\\_confidences — истинные уверенности, pred\\_boxes — предсказанные координаты рамок, target\\_boxes — истинные координаты рамок.

### 3.7 Оптимизатор

В качестве алгоритма оптимизации использовался Adam с параметрами  $lr = 0.001$ . Оптимизатор делает шаг градиентного спуска по следующей формуле:

$$\begin{aligned}m_0 &= 0, v_0 = 0 \\g_t &= \nabla f_t(\theta_{t-1}) \\m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\\hat{m}_t &= \frac{m_t}{(1 - \beta_1^t)} \\\hat{v}_t &= \frac{v_t}{(1 - \beta_2^t)} \\\theta_t &= \theta_{t-1} - lr \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}\end{aligned}$$

где  $\theta_t$  — параметр модели на шаге  $t$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ .

### 3.8 Обучение и валидация

Обучение модели производилось с помощью описанного выше оптимизатора и функции потерь на 100-300 эпохах прохода по батчам.

### 3.9 Методы исследования

Для достижения поставленных целей и задач использовались следующие методы:

- Анализ научно-технической литературы по теме исследования.
- Изменение датасета для более удобной работы с ним.
- Разработка и тестирование архитектур нейронных сетей с использованием фреймворка PyTorch.
- Проведение экспериментов с различными конфигурациями сетей и аугментациями данных.
- Сравнительный анализ результатов моделей по ключевым метрикам.

### 3.10 Научно-теоретическая и практическая значимость исследования

Научно-теоретическая значимость работы заключается в разработке новой архитектуры нейронной сети, интегрирующей свёрточные и рекуррентные компоненты для повышения точности и эффективности детекции объектов на видеопоследовательностях. Практическая значимость исследования состоит в создании системы, способной

применяться в реальных условиях для задач видеонаблюдения, детекции документов и других объектов, что позволяет значительно улучшить качество и надежность этих систем.

## 4 Изменение датасета

Модель YOLO предъявляет требования к формату аннотаций объектов на изображениях, согласно которым необходимо предоставлять четыре числовых значения: координаты центра, ширину и высоту ограничивающей рамки. Все значения должны быть нормированы относительно размеров изображения. В соответствии с данными требованиями, датасет был преобразован следующим образом:

- Файл с аннотациями для каждого изображения содержит только 4 чисела: координаты центра ( $x, y$ ), ширина, высота.
- Все изображения помещены в одну папку, аннотации - в другую.
- Размер изображений уменьшен до 224x224

Примеры изображений с ограничивающими рамками:

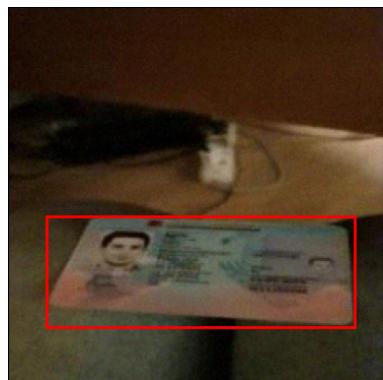


Рисунок 4.1 – Примеры изображений с ограничивающими рамками из преобразованного датасета

## 5 Базовая модель

Была реализована архитектура YOLOv3 с использованием PyTorch:

- Написан Python класс для архитектуры сети.
- Разработан класс для загрузки и обработки датасета в PyTorch.
- Проведено прототипирование основных компонентов сети.

Таблица 5.1 – Архитектура нейронной сети YOLOv3

#	Слой	Параметры	Функция активации	Размер выхода
0	Input	-	-	$3 \times 224 \times 224$
1	Conv	12 фильтров $5 \times 5$ , шаг $1 \times 1$ , отступ $2 \times 2$	SymReLU	$12 \times 224 \times 224$
2	MaxPool	$2 \times 2$ , шаг $2 \times 2$	-	$12 \times 112 \times 112$
3	Conv	16 фильтров $5 \times 5$ , шаг $2 \times 2$ , отступ $1 \times 1$	SymReLU	$16 \times 56 \times 56$
4	Conv	16 фильтров $3 \times 3$ , шаг $2 \times 2$ , отступ $1 \times 1$	SymReLU	$16 \times 28 \times 28$
5	Conv	16 фильтров $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$16 \times 28 \times 28$
6	MaxPool	$2 \times 2$ , шаг $2 \times 2$	-	$16 \times 14 \times 14$
7	Conv	24 фильтра $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$24 \times 14 \times 14$
8	Conv	48 фильтров $3 \times 3$ , шаг $2 \times 2$ , отступ $1 \times 1$	SymReLU	$48 \times 7 \times 7$
9	Conv	48 фильтров $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$48 \times 7 \times 7$
10	Conv	48 фильтров $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$48 \times 7 \times 7$
11	Conv	48 фильтров $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$48 \times 7 \times 7$
12	Conv	64 фильтра $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$64 \times 7 \times 7$
13	Conv	64 фильтра $7 \times 7$ , шаг $7 \times 7$ , отступ $0 \times 0$	SymReLU	$64 \times 1 \times 1$
14	Conv	4 фильтра $1 \times 1$ , шаг $1 \times 1$ , отступ $0 \times 0$	SymReLU	$4 \times 1 \times 1$
15	Conv	4 фильтра $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$4 \times 1 \times 1$
16	Conv	4 фильтра $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$4 \times 1 \times 1$
17	Upsample	scale_factor=(1, 1), mode='nearest'	-	$4 \times 1 \times 1$
18	Conv	96 фильтров $3 \times 3$ , шаг $1 \times 1$ , отступ $1 \times 1$	SymReLU	$96 \times 1 \times 1$
19	Conv	256 фильтров $1 \times 1$ , шаг $1 \times 1$ , отступ $0 \times 0$	SymReLU	$256 \times 1 \times 1$
20	Conv	4 фильтра $1 \times 1$ , шаг $1 \times 1$ , отступ $0 \times 0$	-	$4 \times 1 \times 1$
21	Flatten	start_dim=1	-	4

Свёртка (convolution) является основным блоком свёрточных нейронных сетей. Операция свёртки применяет фильтр (также называемый ядром) ко входному изображению для извлечения признаков. Фильтр скользит по изображению, выполняя элементное умножение и суммирование, создавая карту признаков.

На примере рисунка 5.1, свёртка применяется к входному изображению (слева), используя фильтр  $3 \times 3$ , чтобы создать выходное изображение (справа). На рисунке также добавлены нули по краям входного изображения - это называется отступом.

Макс. пулинг (max pooling) используется для уменьшения размерности карты признаков и сокращения вычислительных затрат, а также для предотвращения переобучения. Эта операция выбирает максимальное значение в каждом подмассиве фиксированного размера.

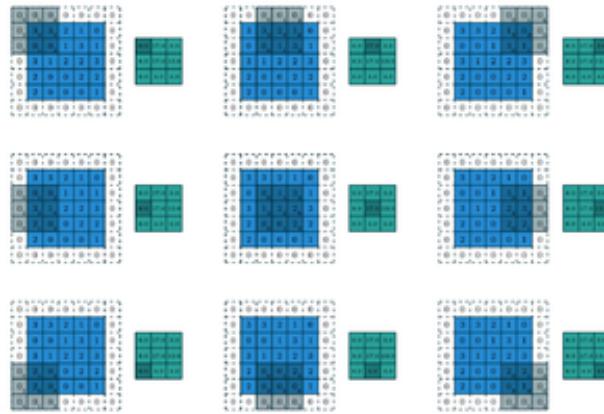


Рисунок 5.1 – Пример операций свёртки

Увеличение (upsample) используется, наоборот, для увеличения размерности карты признаков. Один из наиболее распространенных методов увеличения — это ближайший сосед (nearest neighbor), который дублирует ближайшие значения для увеличения размера. Метод ближайшего соседа используется в предложенной архитектуре.

Такая архитектура сети довольно легкая, что позволяет быстро обучать модели и проводить эксперименты. Она также позволяет работать на ЦПУ на слабых устройствах. Для функции активации была выбрана Symmetrical ReLU (SymReLU)

$$\text{SymReLU}(x) = \max(0, x) - \alpha \min(0, x)$$

где  $\alpha$  — параметр, определяющий влияние отрицательных значений.

В отличие от стандартной ReLU, которая зануляет все отрицательные значения, SymReLU позволяет учитывать отрицательные значения.

## 5.1 Результаты

Модель YOLOv3 была настроена и обучена на подготовленных данных. Получены первичные результаты, демонстрирующие нетривиальные показатели качества. IoU получился 0.77

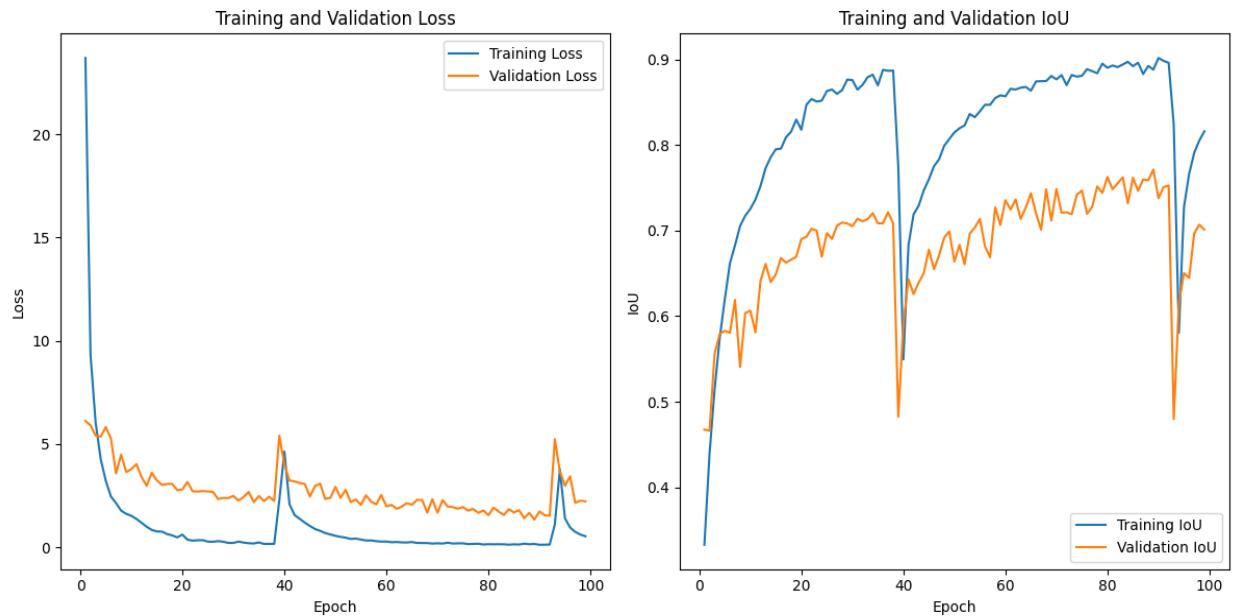


Рисунок 5.2 – Значения функции потерь и качества.



Рисунок 5.3 – Пример работы базовой архитектуры.

## 6 Реализация алгоритмического трекинга

Для улучшения анализа последовательных кадров был разработан алгоритм трекинга, который использует дополнительную метрику - уверенность модели в предсказании. Помимо координат ограничивающей рамки, на вход модели подается еще одно значение - уверенность, которая изначально установлена равной 1, так как объект всегда присутствует на изображении.

В процессе обучения модель подбирает веса так, чтобы уверенность на выходе была максимально близка к 1. Это означает, что модель уверена в правильности своего предсказания. Однако, если на изображении появляется что-то необычное, с чем модель ранее редко сталкивалась, уровень уверенности снижается. Таким образом, уверенность служит индикатором надежности предсказаний модели: чем ниже уверенность, тем меньше модель уверена в правильности своих предсказаний.

Если уверенность меньше определенного порога, то алгоритм выдает ответ такой же, как для предыдущего кадра. Работаем в предположении, что объект за 1 кадр переместился не очень сильно.

### 6.1 Результаты

IoU на валидации составил 0.76.

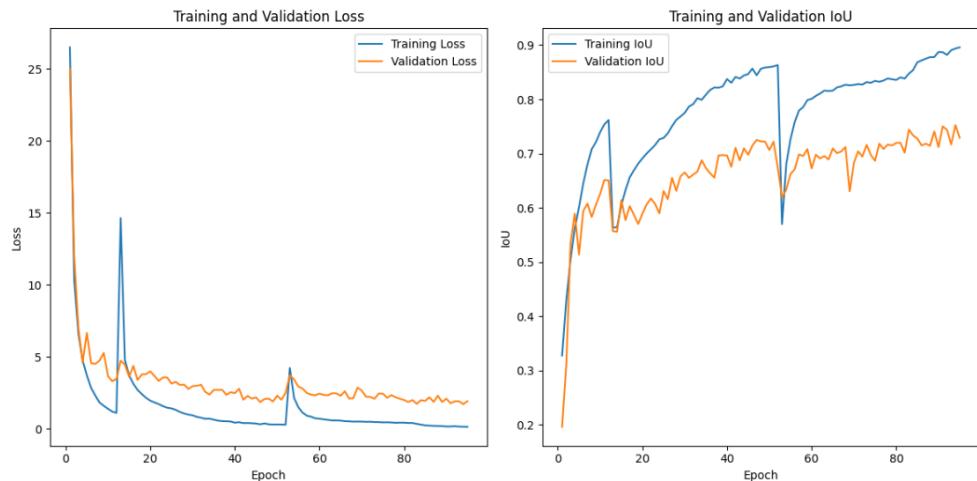


Рисунок 6.1 – Результаты в задаче с алгоритмическим трекингом.

## 7 Разработка рекуррентной YOLO

На данном этапе были внесены изменения в архитектуру нейронной сети. Был добавлен рекуррентный слой GRU перед полносвязным слоем. В результате этих изменений сеть обрабатывает пакет данных, состоящий из 8 последовательных кадров видео, вместо обработки каждого изображения по отдельности.

### 7.1 Изменение структуры датасета

Ранее все изображения хранились в одной папке. Теперь появилась необходимость брать последовательные 8 изображений. Поэтому датасет был преобразован следующим образом:

- Для каждой последовательности из 8 подряд идущих изображений создается отдельная папка.
- Каждый батч содержит 16 таких папок, а минибатч представляет собой одну папку, содержащую 8 изображений.

#### 7.1.1 Преимущества изменений

- **Последовательная обработка кадров:** Сеть обрабатывает минибатчи, состоящие из последовательных кадров.
- **Учет временных зависимостей:** Учитываются временные зависимости между кадрами, что улучшает качество детекции объектов в видео.
- **Перемешивание данных:** Правильная переработка датасета позволяет использовать перемешивание данных при их загрузке в батчи, что улучшает процесс обучения модели, так как она видит изображения в случайном порядке в рамках каждого батча.

Таблица 7.1 – Архитектура нейронной сети с рекуррентным слоем

#	Слой	Параметры	Функция активации	Размер выхода
0	Input	-	-	$8 \times 3 \times 224 \times 224$
1	Conv	12 фильтров 5x5, шаг 1x1, отступ 2x2	SymReLU	$8 \times 12 \times 224 \times 224$
2	MaxPool	2x2, шаг 2x2	-	$8 \times 12 \times 112 \times 112$
3	Conv	16 фильтров 5x5, шаг 2x2, отступ 1x1	SymReLU	$8 \times 16 \times 56 \times 56$
4	Conv	16 фильтров 3x3, шаг 2x2, отступ 1x1	SymReLU	$8 \times 16 \times 28 \times 28$
5	Conv	16 фильтров 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 16 \times 28 \times 28$
6	MaxPool	2x2, шаг 2x2	-	$8 \times 16 \times 14 \times 14$
7	Conv	24 фильтра 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 24 \times 14 \times 14$
8	Conv	48 фильтров 3x3, шаг 2x2, отступ 1x1	SymReLU	$8 \times 48 \times 7 \times 7$
9	Conv	48 фильтров 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 48 \times 7 \times 7$
10	Conv	48 фильтров 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 48 \times 7 \times 7$
11	Conv	48 фильтров 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 48 \times 7 \times 7$
12	Conv	64 фильтра 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 64 \times 7 \times 7$
13	Conv	64 фильтра 7x7, шаг 7x7, отступ 0x0	SymReLU	$8 \times 64 \times 1 \times 1$
14	Conv	4 фильтра 1x1, шаг 1x1, отступ 0x0	SymReLU	$8 \times 4 \times 1 \times 1$
15	Conv	4 фильтра 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 4 \times 1 \times 1$
16	Conv	4 фильтра 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 4 \times 1 \times 1$
17	Upsample	scale_factor=(1, 1), mode='nearest'	-	$8 \times 4 \times 1 \times 1$
18	Conv	96 фильтров 3x3, шаг 1x1, отступ 1x1	SymReLU	$8 \times 96 \times 1 \times 1$
19	Conv	512 фильтров 1x1, шаг 1x1, отступ 0x0	SymReLU	$8 \times 512 \times 1 \times 1$
20	Conv	512 фильтров 1x1, шаг 1x1, отступ 0x0	-	$8 \times 512 \times 1 \times 1$
21	Flatten	start_dim=1	-	$8 \times 512$
22	GRU	input_size=512, hidden_size=512, num_layers=1	-	$1 \times 8 \times 512$
23	FC	512 -> 5	-	$1 \times 8 \times 5$

## 7.2 Аугментации данных

Для повышения точности модели были добавлены различные аугментации данных: размытие, шум и повороты изображений на 0, 90 или 180 градусов. На другие углы повороты не производились. Ниже предоставлен пример изображения с аугментациями.

Модель была обучена на этих данных. Было замечено, что для модели было сложнее детектировать объект на изображениях, повернутых на 90 градусов:

В результате были сохранены лишь повороты на 0 и 180 градусов.

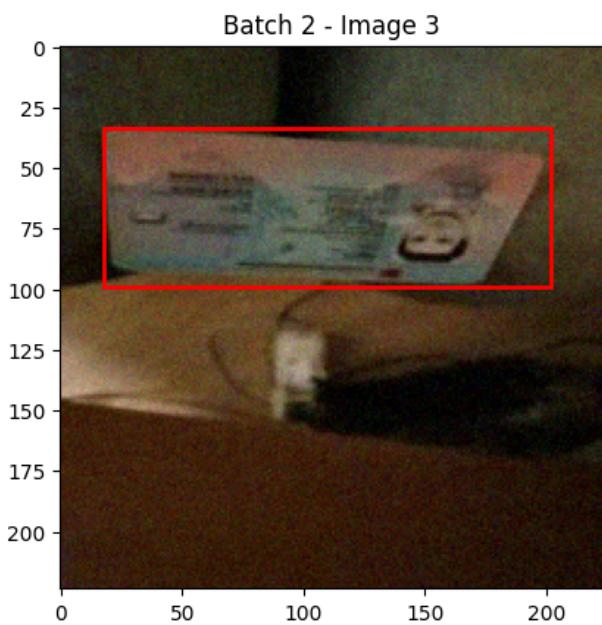


Рисунок 7.1 – Пример изображения с аугментациями

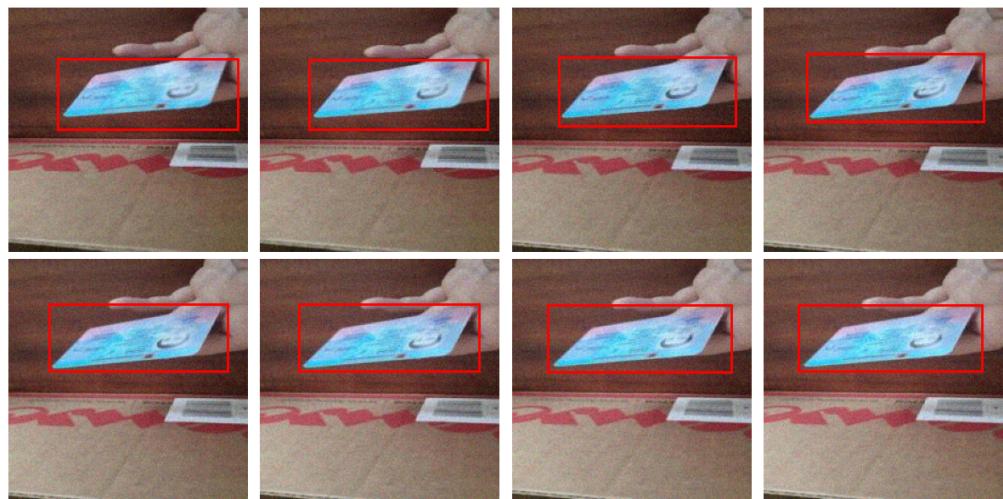


Рисунок 7.2 – Ответ модели для горизонтального документа.

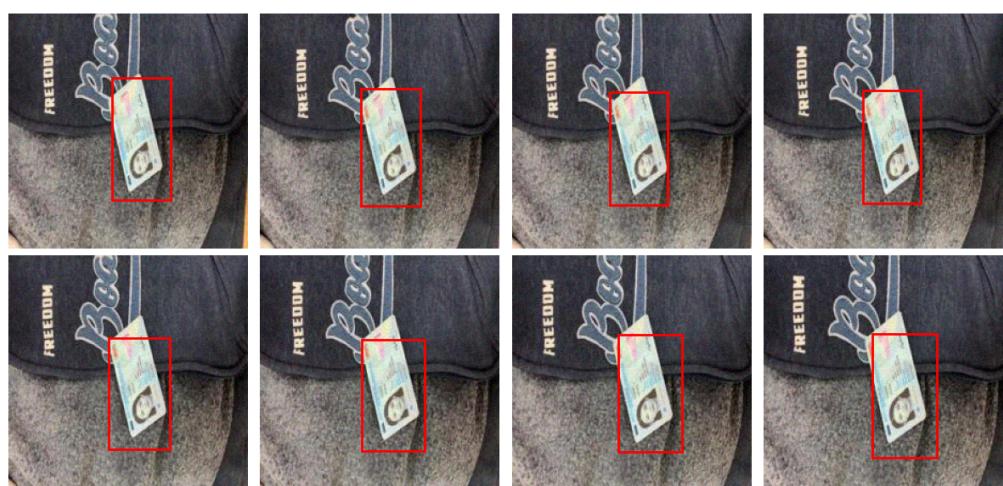


Рисунок 7.3 – Ответ модели для вертикального документа.

### 7.3 Результаты

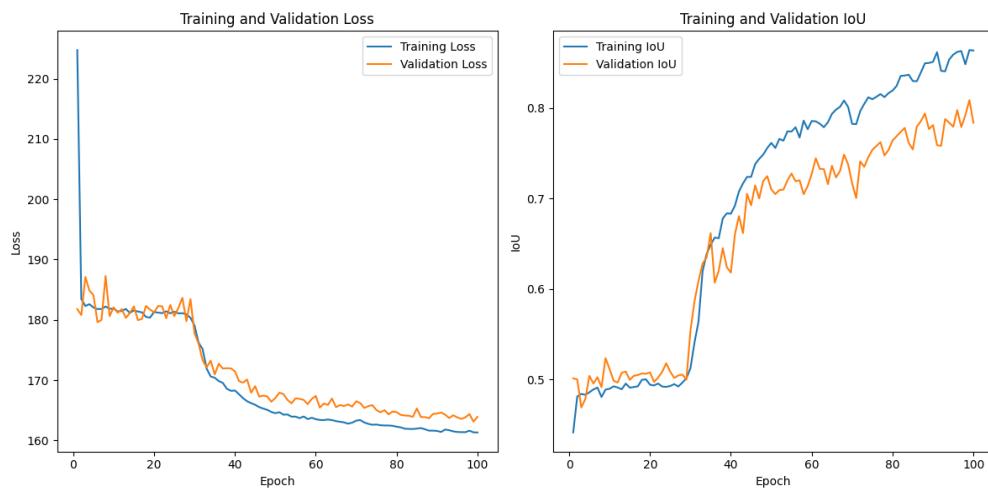


Рисунок 7.4 – Результаты модели с рекуррентным слоем.

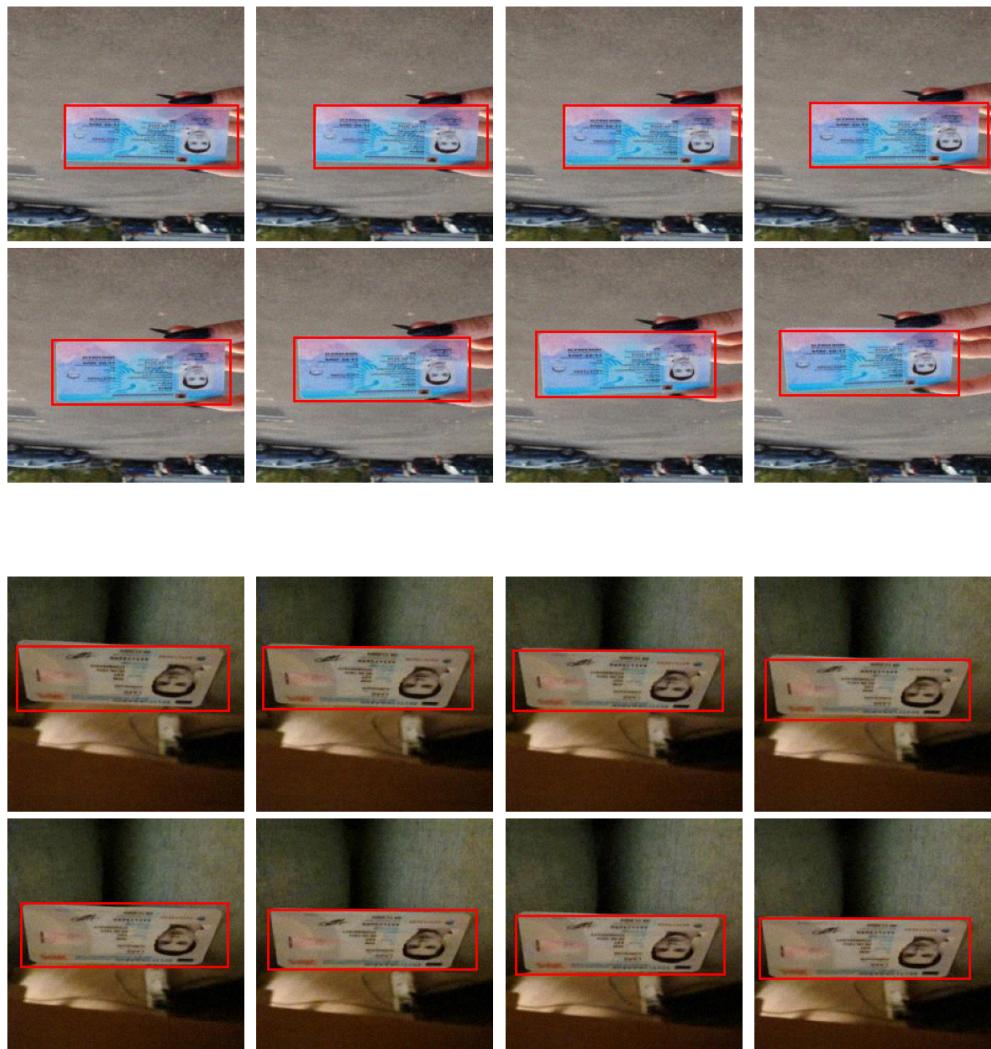


Рисунок 7.5 – Примеры работы модели с рекуррентным слоем.

## 8 Эксперименты

Датасет был разделен на тренировочную и валидационную части неправильно: первые 80% изображений ушли в тренировочную часть, а оставшиеся 20% - в валидационную. В итоге валидационная часть оказалась составленной из изображений с сильными проективными искажениями и бликами от солнца или лампы, поскольку такие изображения находятся в конце датасета.

Позже было выполнено правильное разделение: 80% изображений из всех категорий попали в тренировочную часть, а оставшиеся 20% - в валидационную. Результаты неправильного разбиения также могут быть полезны, так как в реальных условиях иногда невозможно получить идеальный датасет, например, когда изображения генерируются искусственно или когда дизайн объекта меняется.

### 8.1 Неправильное разделение датасета

#### 8.1.1 С аугментациями

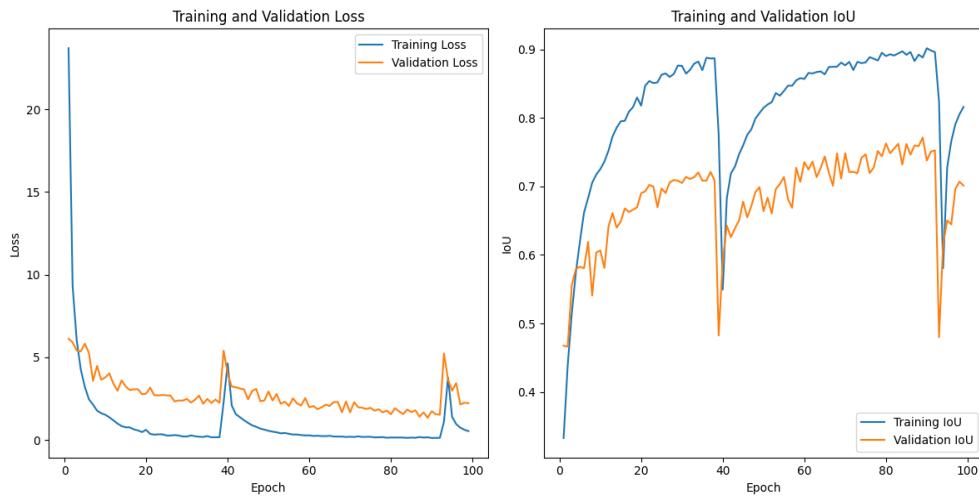


Рисунок 8.1 – Базовая архитектура.

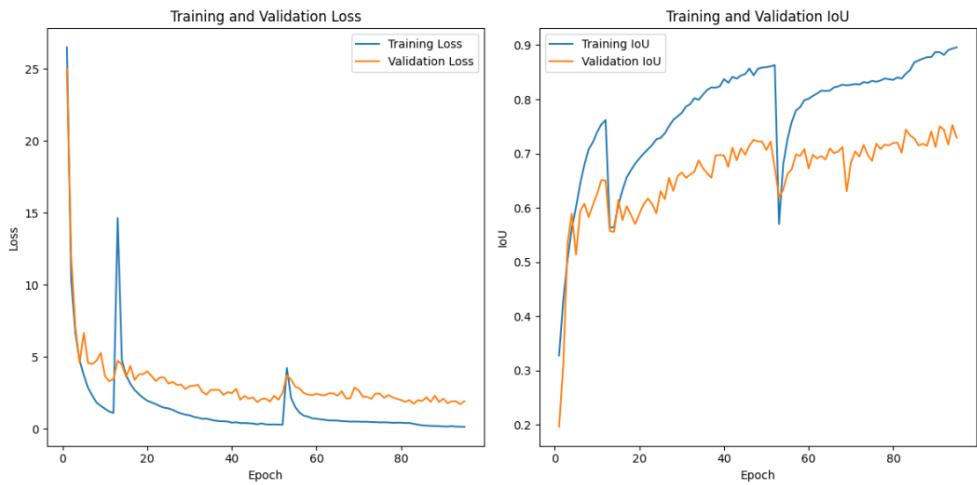


Рисунок 8.2 – Архитектура с алгоритмическим трекингом.

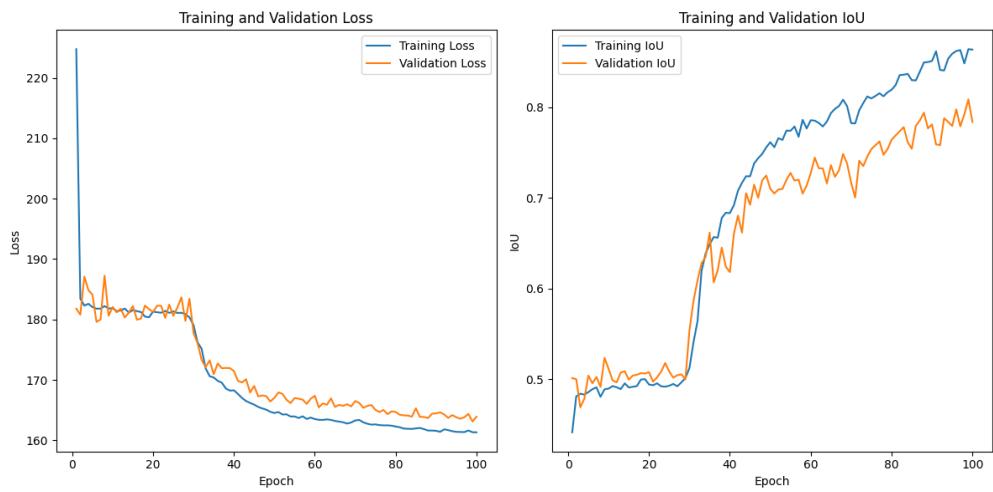
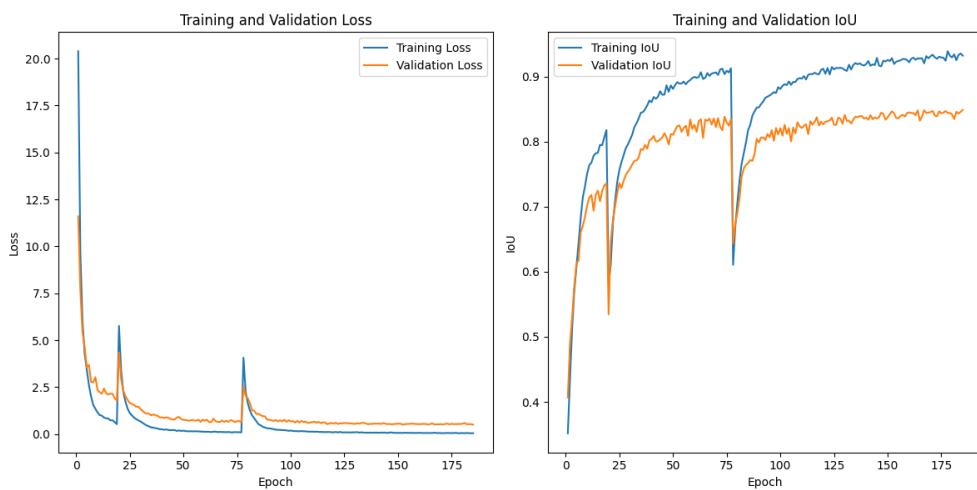


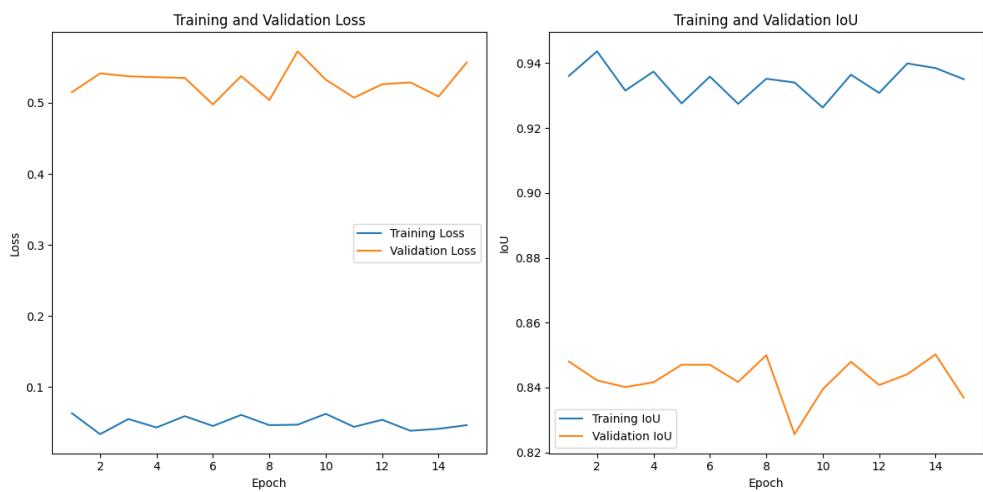
Рисунок 8.3 – Архитектура с рекуррентным слоем.

## 8.2 Правильное разделение датасета

### 8.2.1 Без аугментаций



(a) Базовая архитектура - 185 эпох



(b) Базовая архитектура - последние 15 эпох.

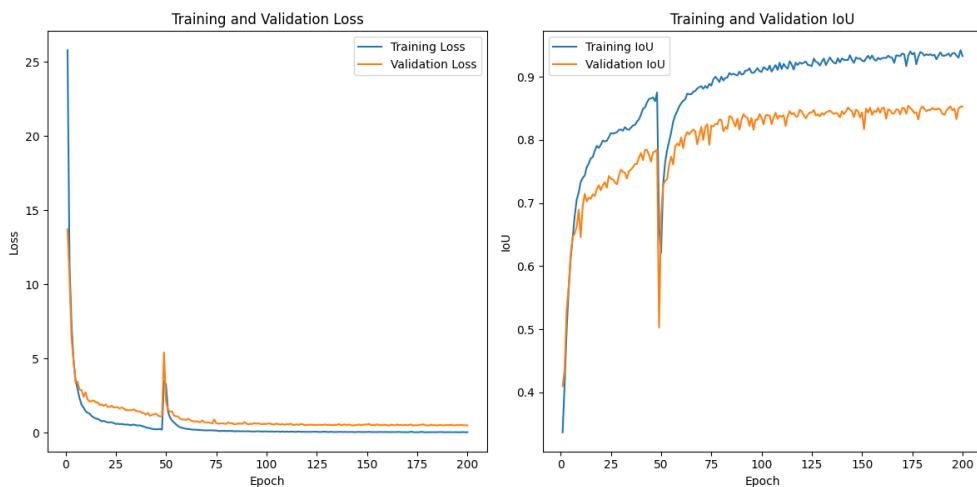


Рисунок 8.5 – Архитектура с алгоритмическим трекингом.

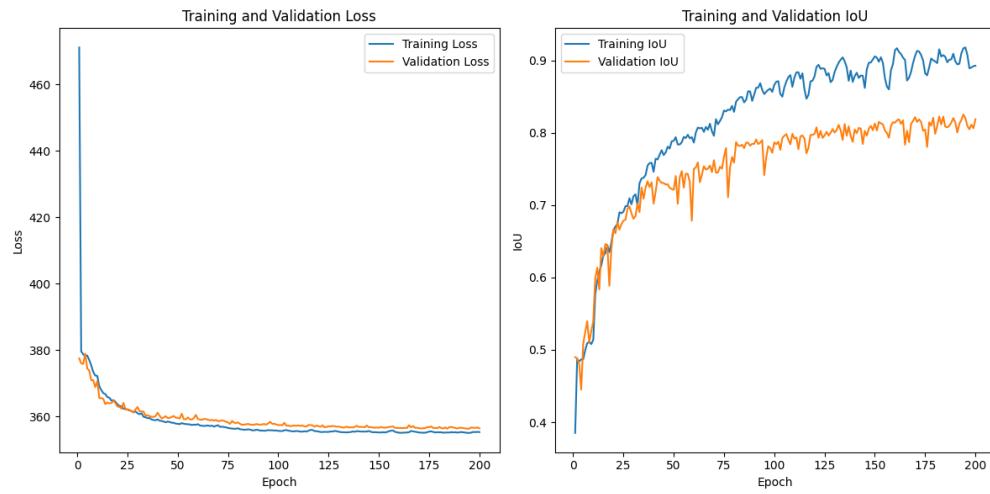


Рисунок 8.6 – Архитектура с рекуррентным слоем.

## 8.2.2 С аугментациями

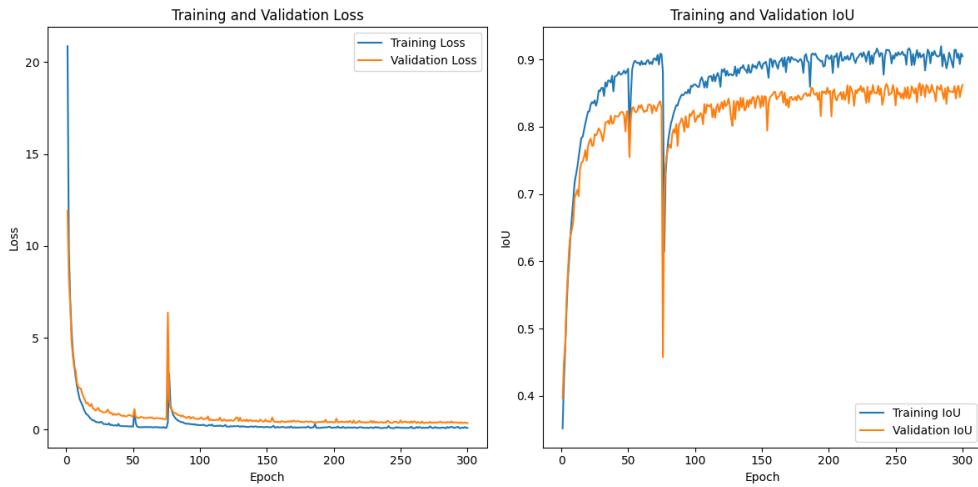


Рисунок 8.7 – Базовая архитектура.

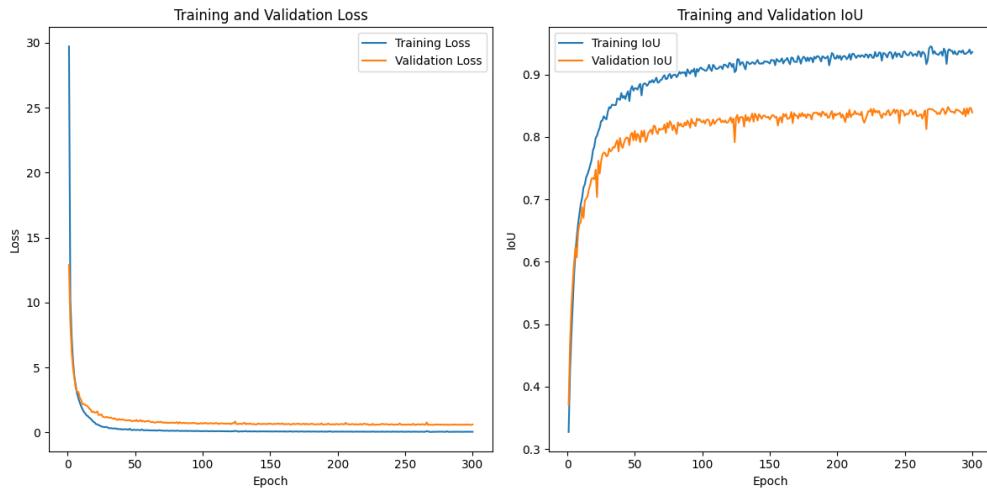


Рисунок 8.8 – Архитектура с алгоритмическим трекингом.

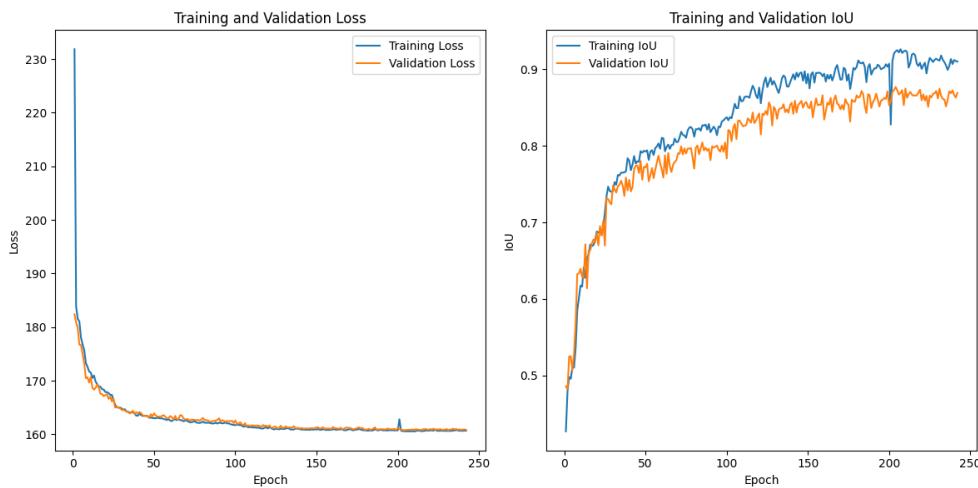


Рисунок 8.9 – Архитектура с рекуррентным слоем.

### 8.3 Общие результаты

Таблица 8.1 – Значения IoU для различных архитектур

	Неправильное разделение	Правильное разделение	
	С аугментациями	Без аугментаций	С аугментациями
Базовая	0.77	0.85	0.86
Трекинг	0.76	0.85	0.85
Рекуррентная	0.81	0.83	0.88

Можно также посмотреть результаты на конкретных примерах. Ниже предоставлены примеры работы базовой модели и модели с рекуррентным слоем при наличии аугментаций и правильном разделении датасета.

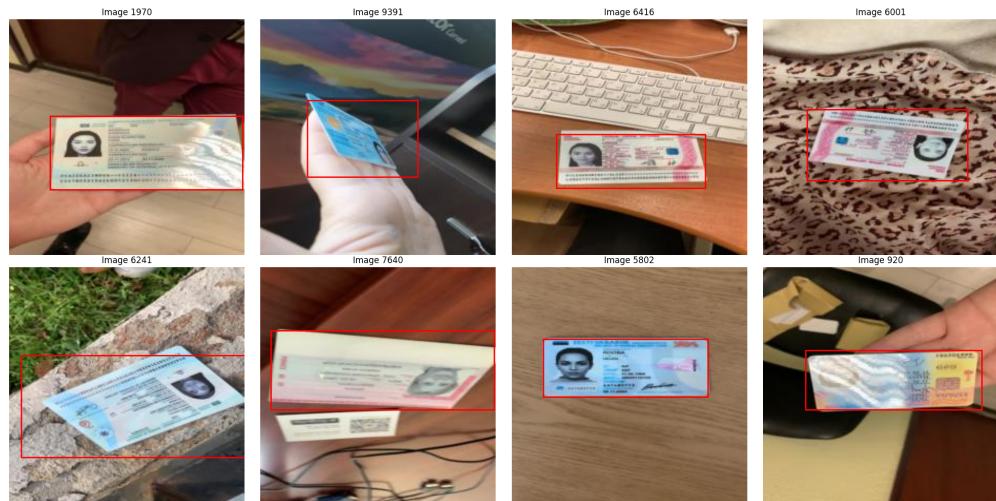


Рисунок 8.10 – Пример работы базовой архитектуры

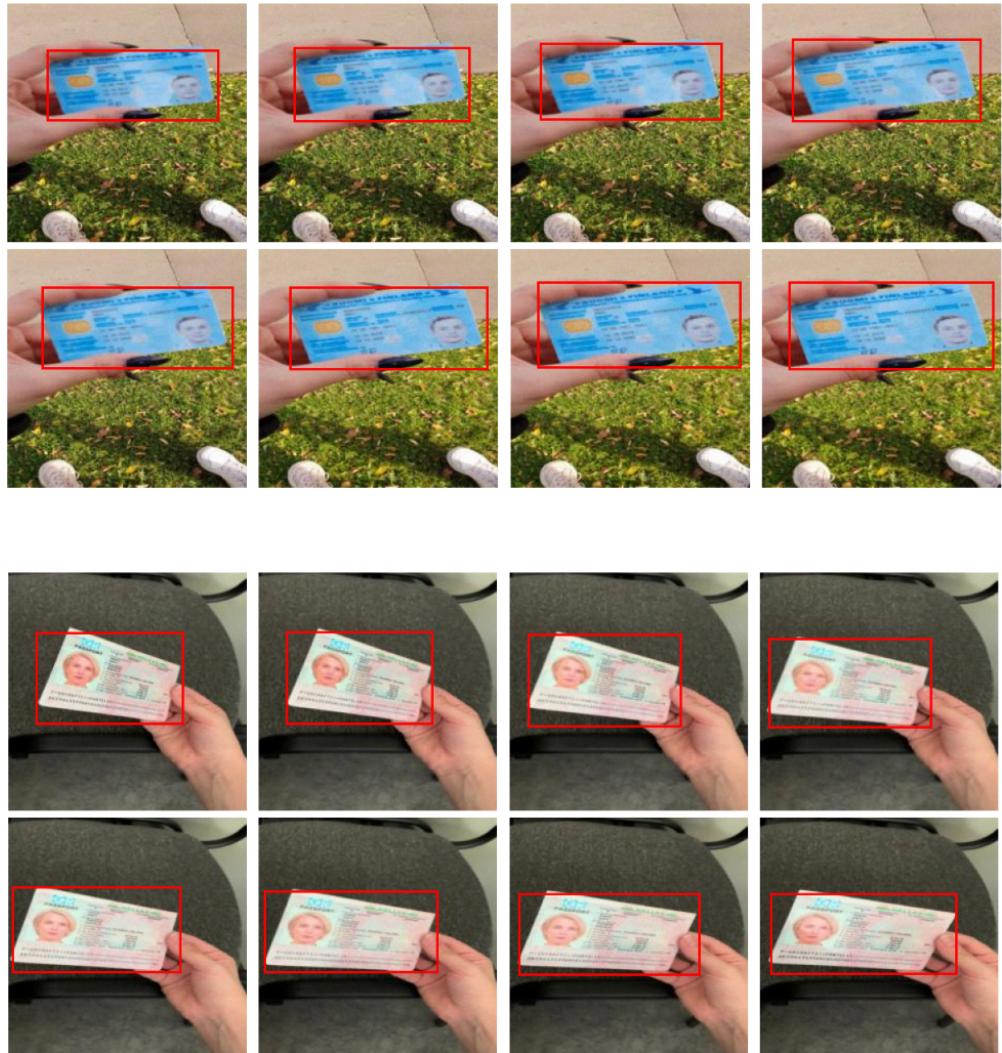


Рисунок 8.11 – Примеры работы модели с рекуррентным слоем.

## 9 Заключение

В рамках дипломной работы были выполнены следующие задачи:

1. Проведен детальный анализ литературы, охватывающий современные методы детекции объектов и использование сверточных и рекуррентных нейронных сетей.
2. Разработана и протестирована архитектура YOLO, продемонстрировавшая высокую точность при детекции объектов.
3. Внедрен алгоритмический трекинг, который считает уверенность модели в своем ответе и выдает ответ для предыдущего кадра, если уверенность низкая.
4. Интегрирован рекуррентный слой в YOLO для учета временных зависимостей между кадрами, что повысило качество детекции.
5. Проведены эксперименты в различных условиях.

6. Рекуррентная архитектура сети показала лучшие результаты по сравнению с другими моделями.

### **Вывод**

В задаче детекции документов на видео при наличии аугментаций добавление рекуррентного слоя в модель YOLO улучшает детекцию.

### **Идеи для дальнейших исследований**

- Провести эксперименты на других датасетах.
- Поэкспериментировать с аугментациями и гиперпараметрами.
- Провести эксперименты с более актуальными моделями YOLO.

## Список литературы

- [1] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You only look once: Unified, real-time object detection // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 779-788.
- [2] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L. Large-scale video classification with convolutional neural networks // Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
- [3] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T. Long-term recurrent convolutional networks for visual recognition and description // arXiv preprint arXiv:1411.4389. 2015.
- [4] Qiao, S., Chen, L. C., Yuille, A. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution // arXiv preprint arXiv:2006.02334. 2020.
- [5] Арлазаров, В.В., Жуковский, А.Е., Кривцов, В.Е., Николаев, Д.П., Полевой, Д.В. Анализ особенностей использования стационарных и мобильных малоразмерных цифровых видеокамер для распознавания документов // Проблемы информационных технологий. 2016. Т. 16, № 2. С. 34-42.
- [6] Арлазаров, В.Л., Кулатов, П.А., Логинов, А.С., Славин, О.А. Алгоритмы поиска границ печатных символов, используемые при оптическом распознавании символов // Проблемы информационных технологий. 2019. Т. 17, № 3. С. 56-67.
- [7] Hartley, R., Zisserman, A. Multiple view geometry in computer vision. Cambridge: Cambridge University Press, 2003.
- [8] Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. YOLOv4: Optimal speed and accuracy of object detection // arXiv preprint arXiv:2004.10934. 2020.
- [9] Zhang, J., Zhang, H., Liu, B., Qu, G., Wang, F., Zhang, H., Shi, X. Small object intelligent detection method based on adaptive recursive feature pyramid // Heliyon. 2023. Vol. 9, No. 5. e09863.
- [10] Smart Engines. Хабр. URL: <https://habr.com/ru/companies/smartengines/articles/714250/> (дата обращения: 15.09.2023).
- [11] K.B. Bulatov, E.V. Emelianova, D.V. Tropin, N.S. Skoryukina, Y.S. Chernyshova, A.V. Sheshkus, S.A. Usilin, Z. Ming, J.-C. Burie, M. M. Luqman, V.V. Arlazarov: “MIDV-2020: A Comprehensive Benchmark Dataset for Identity Document Analysis”, Computer Optics (submitted), 2021.