МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(*Наименование института*)
Кафедра «Прикладная математика и информатика»
(*Наименование кафедры, центра, департамента*)

ОТЧЕТ
по производственной практике
(научно-исследовательской работе) 3

(*наименование практики*)

ОБУЧАЮЩЕГОСЯ А.А. ДЬЯЧЕНКО
(И.О. Фамилия)

НАПРАВЛЕНИЕ ПОДГОТОВКИ (СПЕЦИАЛЬНОСТЬ) 01.04.02 Прикладная математика и информатика

ГРУППА ПМИм-2001а

РУКОВОДИТЕЛЬ ПРАКТИКИ ОТ УНИВЕРСИТЕТА:

Гущина Оксана Михайловна, доцент кафедры, к.п.н.
(фамилия, имя, отчество, должность)

Руководитель практики от организации
(предприятия, учреждения, сообщества)

Талалов Сергей Владимирович, профессор
(фамилия, имя, отчество, должность)

Тольятти 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(*Наименование института*)
Кафедра «Прикладная математика и информатика»
(*Наименование кафедры, центра, департамента*)

АКТ о прохождении практики

Данным актом подтверждается, что

ОБУЧАЮЩИЙСЯ А.А. ДЬЯЧЕНКО
(И.О. Фамилия)

НАПРАВЛЕНИЕ ПОДГОТОВКИ (СПЕЦИАЛЬНОСТЬ) 01.04.02 Прикладная математика и информатика

ГРУППА ПМИм-2001а

Проходил производственную практику
(научно-исследовательскую работу) 3
(*Наименование практики*)

в ФГБОУ ВО Тольяттинский государственный университет
(*Наименование организации*)

в период с_____по _____ г.

Руководитель практики от организации
(предприятия, учреждения, сообщества):
Талалов Сергей Владимирович, профессор
(фамилия, имя, отчество, должность)

РЕКОМЕНДУЕМАЯ ОЦЕНКА __ _____

_____ _____
(дата) (подпись)

М.П.

АННОТАЦИЯ

Работа включает в себя 21 страницу, 8 рисунков, 2 таблицы, и 12 используемых источников.

В данной научной исследовательской работе (НИР) проводится анализ архитектуры YOLOR с целью улучшения точности его работы с минимальным влиянием на скорость обнаружения.

Объектом исследования является современный алгоритм обнаружения множества объектов в режиме реального времени YOLOR.

Предметом исследования является производительность используемых частей детектора и узкие места архитектуры.

Целью данной исследовательской работы является моделирование точного детектора объектов, способного работать в режиме реального времени на основе YOLOR.

В результате выполнения НИР была предложена измененная архитектура детектора YOLOR, которая способна значительно повысить точность модели, сохраняя скорость работы, требуемую для приложений в реальном времени.

Ключевые слова: обнаружение объектов, детектор, алгоритм, реальное время, модель, сверточная нейронная сеть

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Компьютерное зрение широко используется в различных прикладных задачах, что повышает интерес к данному направлению со стороны исследователей. В связи с этим такие направления, как классификация изображений, обнаружение и сегментация объектов быстро развиваются. Периодически происходит повышение скорости и точности выполнения этих задач за счет введения новых методов. Нередки случаи, когда применение тех же методов к другим алгоритмам, либо их комбинирование в рамках одного подхода позволяет значительно повысить производительность улучшаемой модели и скорректировать вектор дальнейших исследований в этом направления.

В задачах обнаружения объектов важны как точность, так и скорость обнаружения, по этой причине одноступенчатые алгоритмы семейства YOLO на протяжении 5 лет держат лидерство в данной области, обеспечивая наилучший баланс между точностью и быстродействием. Самым лучшим методом в данном семействе и среди других алгоритмов для обнаружения объектов в режиме реального времени на данный момент является YOLOR.

Данное исследование направлено на анализ архитектуры YOLOR с целью улучшения точности его работы с минимальным влиянием на скорость обнаружения.

Объектом исследования является современный алгоритм обнаружения множества объектов в режиме реального времени YOLOR.

Предметом исследования является производительность используемых частей детектора и узкие места архитектуры.

В результате выполнения НИР были выявлены слабые места архитектуры детектора YOLOR и даны предложения по улучшению ее точности, сохраняя скорость работы, требуемую для приложений реального времени.

1. Описание архитектуры алгоритма YOLOR

С целью создания оптимального алгоритма обнаружения объектов в режиме реального времени в предыдущих исследованиях был произведен сравнительный анализ современных методов множественного обнаружения объектов на основе сверточных нейронных сетей. Среди рассмотренных алгоритмов наилучшие результаты точности и скорости показал YOLOv4. Относительно недавно, на основе данного алгоритма был разработан подход YOLOR, который на 2021 год является самым точным для приложений реального времени [5].

Основная идея метода YOLOR состоит в совместном использовании явных и неявных знаний для обнаружения объектов. Под явными знаниями подразумеваются те, которые могут быть получены в процессе обучения, а под неявными те, которые получены подсознательно. Проецируя данные понятия в область сверточных нейронных сетей можно сказать, что явные знания это признаки, извлекаемые из начальных слоев, а неявные знания – признаки, полученные из глубоких слоев. Объединяя явные и неявные знания в процессе обнаружения, данный метод получает точность 55.5 AP и скорость 66 fps [5]. Общий принцип работы данного подхода представлен на рисунке 1.1.

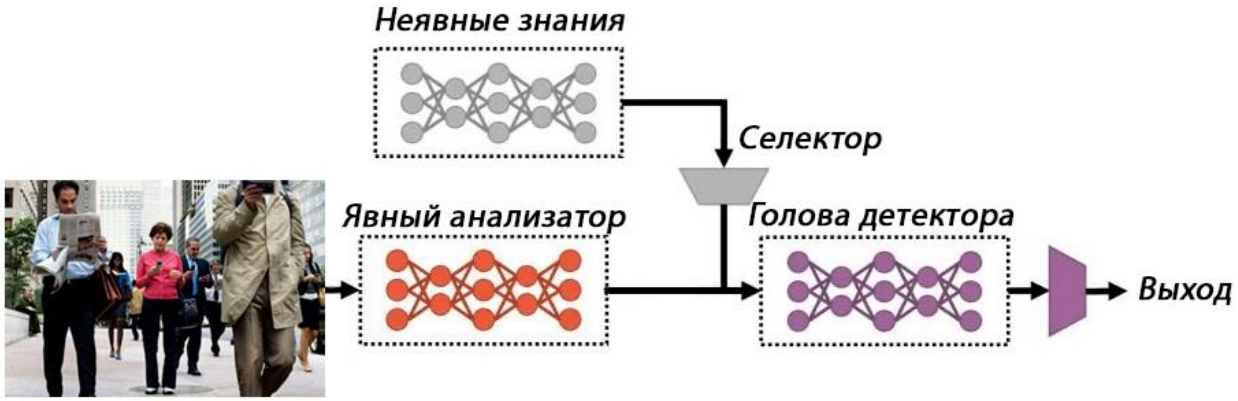

Рисунок 1.1 – Схема метода YOLOR

Явное обучение может производиться разными способами, в зависимости от решаемой задачи, но в случае с обнаружением объектов под явным обучением подразумевается обычное обучение фрагмента сети,

отвечающего за получение карт признаков, с помощью обратного распространения ошибки для оптимизации его параметров.

Явный анализатор, представленный на рисунке 1.1 представляет из себя экстрактор признаков CSPDarknet-53 и сети пирамид признаков PAN + SPP для уточнения признаков [1]. Архитектура экстрактора признаков CSPDarknet-53 представлена на рисунке 1.2.

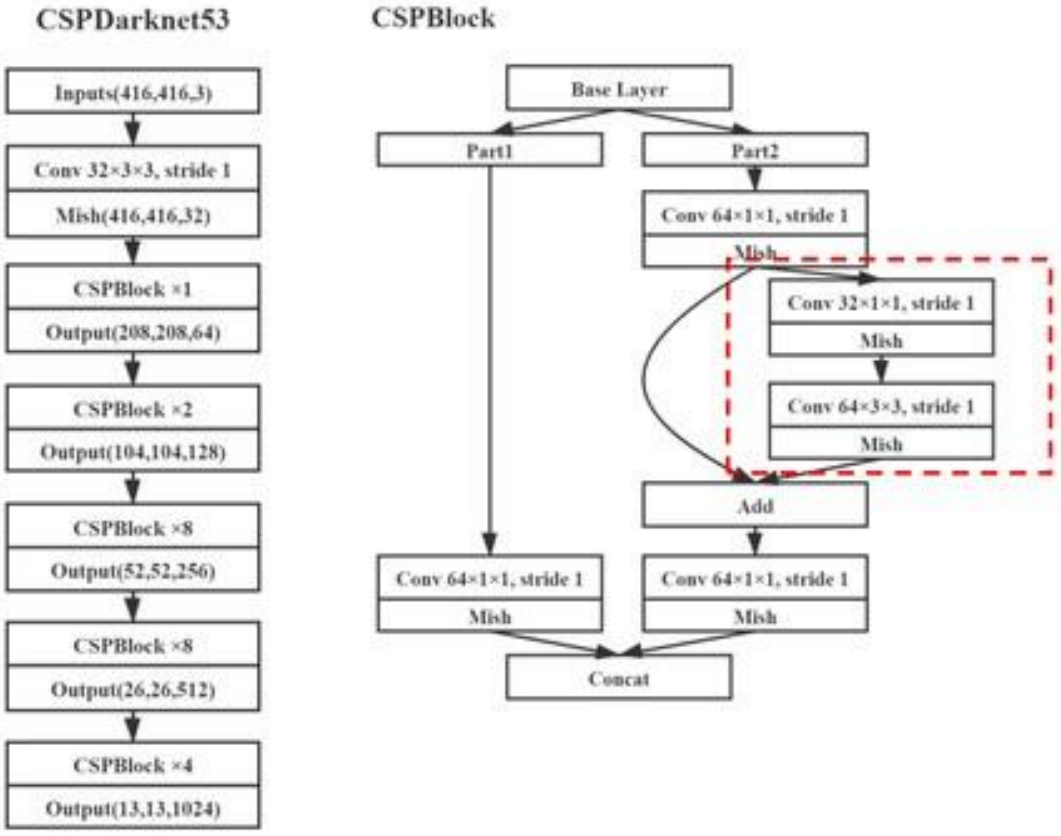

Рисунок 1.2 – Структурная модель CSPDarknet-53

CSPDarknet-53 представляет из себя стандартную архитектуру Darknet-53 с добавлением межэтапных частичных соединений, предложенных в CSPNet. Добавление межэтапных частичных соединений в данную архитектуру, производит усечение градиентного потока, что повышает способность к обучению у модели и устраняет узкие места в вычислениях [4]. Выбор данного экстрактора авторами обуславливается тем, что данная архитектура демонстрирует большую способность повышать точность детектора благодаря различным улучшениям [1].

В качестве метода уточнения карт признаков модель YOLOR использует архитектуру сети пирамид признаков PAN. Структурная модель данной сети представлена на рисунке 1.3.

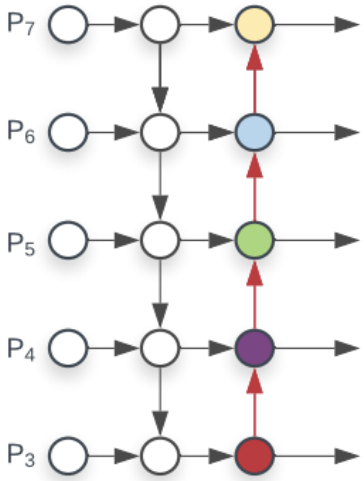

Рисунок 1.3 – Структурная модель PAN

Данная архитектура сети пирамид признаков сокращает информационный путь и усиливает пирамиду признаков точными сигналами локализации с низких уровней посредством добавления к FPN дополнительного восходящего пути [2].

После уточнения карт признаков в методе YOLOR производится объединение пространственных пирамид подобно SPPNet. Авторы обуславливают использование данной операции необходимостью расширения рецептивного поля [6].

Детектор в виде нескольких голов, представляющие сверточные блоки производит прогнозы в трех различных масштабах. Карты признаков с разных уровней PAN покрываются фиксированным набором ограничивающих прямоугольников, размер которых рассчитывается с помощью алгоритма классификации K-NN. Для каждого ограничивающего прямоугольника с помощью сверточных слоев регрессии и классификации предсказываются смещения по координатам для ограничивающей рамки объекта, оценка объективности и распределение вероятностей по классам. Далее с помощью

алгоритма подавления не-максимумов удаляются лишние ограничивающие рамки [1].

Неявные знания, применяемые к явным знаниям и представленные на рисунке 1.1 могут быть представлены одни из следующих способов:

- вектор, матрица или тензор;
- нейронная сеть, где неявное представление представляет из себя линейную комбинацию вектора и весовой матрицы;
- матричная факторизация, где неявное представление представлено в виде комбинации базиса нескольких векторов и некоторого коэффициента;

Для получения неявных знаний используют неявные нейронные представления, которые позволяют отображать дискретные входные значения на параметризованное непрерывное множество. Для преобразования неявного обучения в остаточную форму нейронных сетей используются равновесные модели [5].

При введении неявных знаний в модель можно получить следующие преимущества:

- уменьшение пространства многообразия обученного представления, что увеличивает эффективность классификации;
- выравнивание признаков больших и мелких объектов в сетях пирамид признаков;
- автоматический поиск набора гиперпараметров для якорных блоков.

В модели YOLOR неявные знания применяются к выходам PAN для выравнивания признаков, к результирующим предсказаниям детектора для их уточнения и для каждой головы детектора для улучшения оптимизации функции потерь [5]. Применение неявных знаний к модели YOLOR представлено на рисунке 1.4.

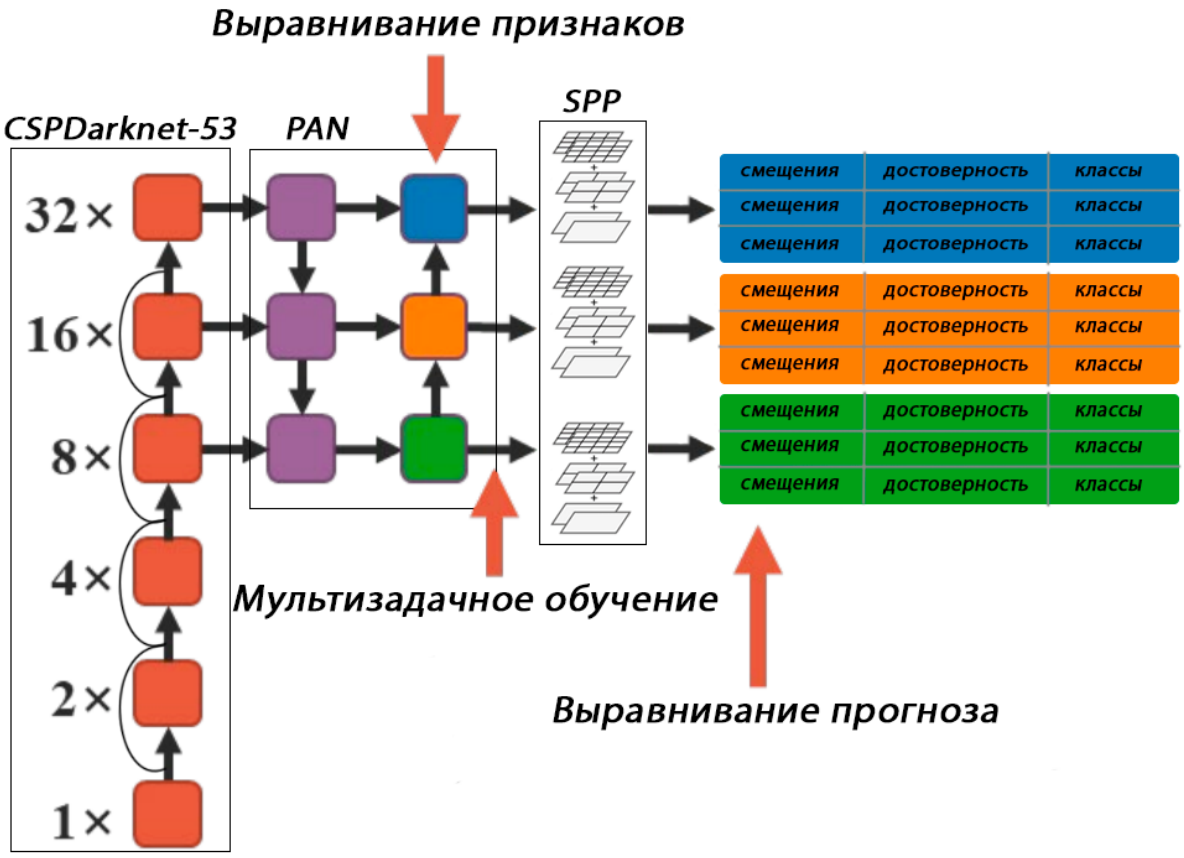

Рисунок 1.4 – Применение неявных знаний к модели YOLOR

Таким образом, в данном разделе были описаны принцип работы и дизайн сети алгоритма YOLOR. Данный алгоритм показывает отличную скорость и достаточно хорошую точность для множественного обнаружения объектов в приложениях реального времени. При анализе архитектуры YOLOR и сравнении его с другими современными детекторами были обнаружены возможности улучшения конструкции модели, способные повысить производительность детектора.

2. Предлагаемые улучшения

2.1 Улучшение экстрактора признаков

Так как точность модели во многом зависит от качества извлечения признаков из исходного изображения, был проведен сравнительный анализ основных экстракторов признаков с целью выявления возможностей по улучшению экстрактора признаков YOLOR.

Среди существующих экстракторов признаков наибольшей популярностью пользуются архитектуры VGG, ResNet, Darknet, DenseNet, DPN.

Архитектура VGG является очень простой, но весьма эффективной за счет подхода наращивания сверточных блоков одинаковой формы.

Архитектура ResNet подобно VGG использует стратегию использования блоков с одинаковой структурой, добавляя к ним остаточные соединения, что позволяет решить проблему исчезновения градиента при наращивании глубины сети [7].

Архитектуры ResNeXt и Res2Net повторяют блочную структуру ResNet, но изменяют структуру блоков.

В архитектуре сети ResNeXt вместо последовательных сверток в блоке используются независимые стеки веток, объединяющие результаты. Данная архитектура увеличивает точность, без изменения сложности и количества параметров [9].

В Res2Net строение блока представляет из себя иерархические остаточные соединения. Данная архитектура позволяет более детально изменять рецептивные поля, захватывая локальные и глобальные признаки [10].

Модель Darknet объединяет блочность и простоту VGG и остаточные соединения ResNet [1].

В архитектуре DenseNet внутри каждого из чередующихся блоков карты признаков, получаемые с каждого слоя, поступают на вход всем остальным

слоям. Для их объединения применяется конкатенация. В результате, получается извлекать более разнообразные карты функций.

Архитектура DPN объединяет ResNet с DenseNet, что позволяет обеспечить возможность повторного использования функций и возможность более обширного исследования признаков [12].

В таблице 1 представлено сравнение основных параметров моделей: количество слоев, количество параметров, количество выполняемых операций с плавающей запятой (FLOPs) и точность на наборе данных ImageNet.

Таблица 1. Сравнение параметров экстракторов признаков

Наименование модели	Количество слоев	Количество параметров (млн.)	FLOPs (млрд.)	Точность (Топ-1)
VGG	16	138	15.3	0.713
	19	144	19.6	0.713
ResNet	50	23	3.8	0.758
	101	42	7.6	0.771
	152	58	11.3	0.766
ResNeXt	50	23	3.8	0.774
	101	42	7.6	0.788
	152	58	11.3	0.790
Res2Net	50	23	3.8	0.779
	101	42	7.6	0.792
	152	58	11.3	0.797
Darknet	19	50	7.29	0.729
	53	40	18.7	0.772
DenseNet	121	7.2	5.6	0.750
	169	12.8	6.7	0.762
	201	20	8.6	0.773
	264	33	11.5	0.778
DPN	92	37.8	6.5	0.793
	98	61.7	11.7	0.798

Из результатов сравнения видно, что стандартная реализация Darknet-53 уступает по точности ResNeXt, Res2Net и DPN. В таком случае, использование

одного из данных экстракторов признаков совместно с межступенчатыми частичными соединениями позволит увеличить точность детектора. Исходя из того, что DPN при лучших показаниях точности имеет наибольшее количество параметров и операций с плавающей запятой, ввод данного экстрактора увеличит время обучения и время вывода модели. По этой причине в качестве нового экстрактора признаков было решено использовать новый экстрактор признаков на основе ResNet-101, с использованием блоков ResNeXt и Res2Net. Данная архитектура позволит объединить преимущества этих подходов [10]. Модули ResNeXt и Res2Net в сравнении с ResNet представлены на рисунке 2.1.

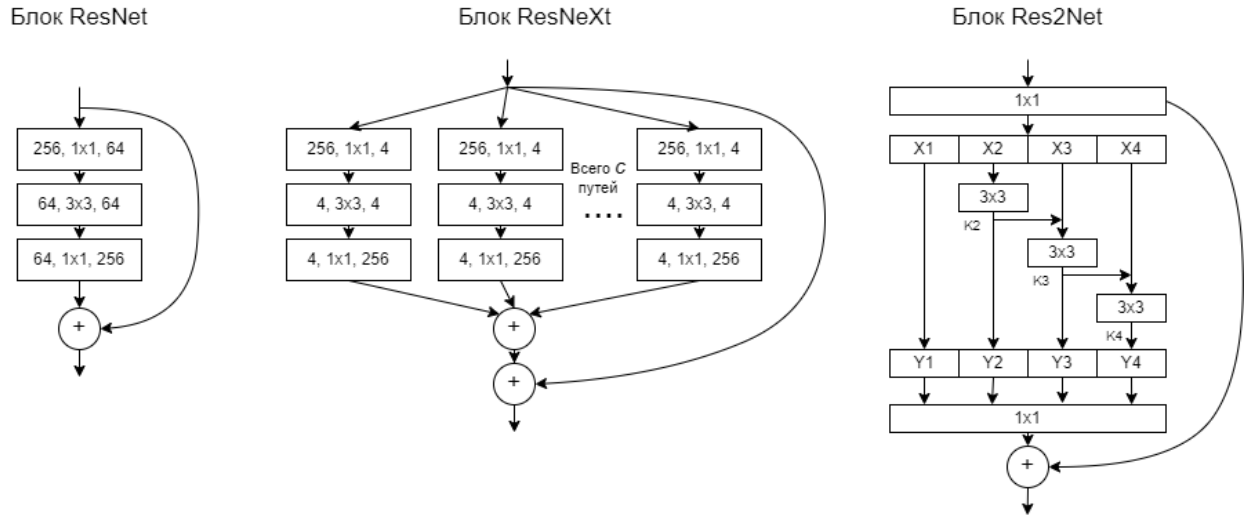

Рисунок 2.1 – Строение блоков ResNeXt и Res2Net

Использование предложенной в ResNeXt концепции «сеть в нейроне» позволяет разбивать карты признаков на несколько наборов для произведения независимых сверток, в дальнейшем объединяя их. Вводится новое измерение, называемое «мощность» регулирующее количество таких независимых ветвей сверток [9]. Подобная архитектура экстрактора на практике показывает значительное увеличение разнообразности признаков и, следовательно, точности классификации, без увеличения количества параметров и сложности сети.

Инкапсуляция модуля Res2Net с иерархическими остаточными соединениями позволяет изменять рецептивные поля на более детальном уровне для захвата мелких деталей и глобальных признаков. Данная

архитектура также вводит новое измерение, называемое «масштаб», регулируемое количество групп признаков в блоке [10].

Общая структура предложенного экстрактора признаков представлена на рисунке 2.2.

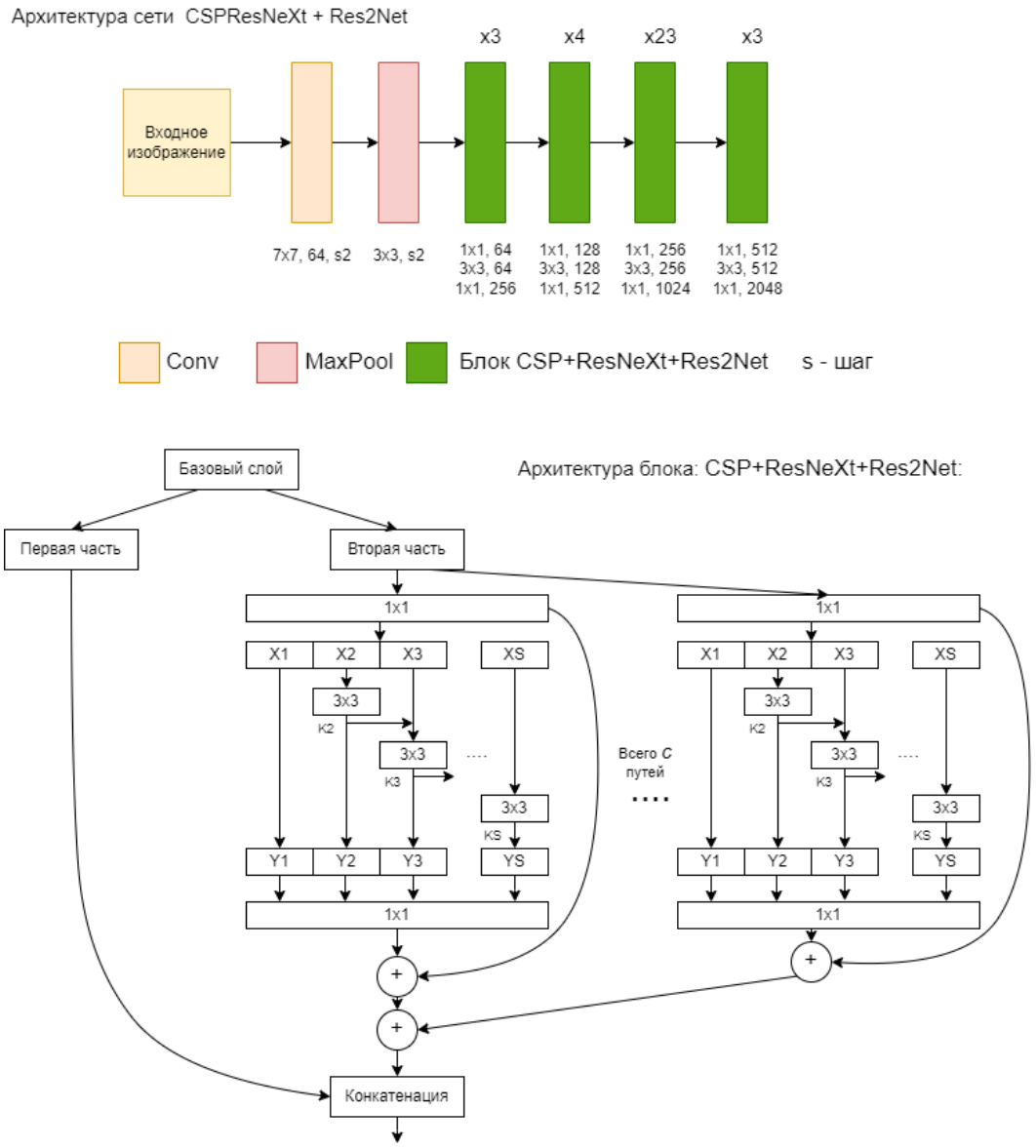

Рисунок 2.2 – Архитектура предлагаемого экстрактора признаков

Выбор именно 101-слойной архитектуры вместо 50-слойной обуславливается увеличением рецептивного поля при увеличении количества слоев, что способствует обнаружения изменений в более широкой области [7]. Архитектура с 152 слоями дает небольшой прирост к точности, но может значительно увеличить скорость работы детектора, что является нежелательным.

2.2. Улучшение метода уточнения карт признаков

Уточнение карт признаков является не менее важным этапом, чем их извлечение. Правильное конструирование шеи детектора может значительно повысить производительность модели. Почти все методы уточнения карт признаков основаны на сетях пирамид признаков и используют восходящие и нисходящие преобразования карт признаков с одноуровневыми соединениями [11].

С целью улучшения сети уточнения карт признаков был проведен сравнительный анализ таких сетей, как FPN, PAN, NAS-FPN, ASFF, BiFPN и TPN.

FPN (Feature Pyramid Networks) объединяет семантически сильные признаки низкого разрешения с семантически слабыми признаками высокого разрешения посредством нисходящего пути и боковых связей [7].

PAN (Path Aggregation Network) сокращает информационный путь и усиливает пирамиду признаков точными сигналами локализации с низких уровней посредством добавления к FPN восходящего расширения [2].

NAS-FPN (Neural Architecture Search - FPN) позволяет сети самостоятельно конструировать свою архитектуру для нахождения оптимальных соединений в заданном пространстве поиска.

ASFF (Adaptively Spatial Feature Fusion) позволяет решать проблему несогласованности объединяемых данных с помощью идентичного изменения масштаба и адаптивного объединения.

BiFPN (Bi-directional Feature Pyramid Network) обеспечивает более высокоуровневое объединение признаков за счет использования взвешенной двунаправленной пирамидальной сети признаков [8].

TPN (Trident Pyramid Networks) производит объединение признаков с помощью обработок на основе связи подобно BiFPN и самообработок, напоминающих структуру блока с остаточными соединениями [3].

Результаты сравнения описанных архитектур уточнения признаков приведены в таблице 2.

Таблица 2. Сравнение архитектур уточнения признаков

Наименование архитектуры уточнения признаков	Увеличение параметров (млн.)	Среднее повышение точности (%)
FPN	8	8
PAN	15	11
NAS-FPN	37.3	10
ASFF	13	10.3
BiFPN	11.7	12.5
TPN	14.1	14

Из приведенной таблицы видно, что метод PAN, используемый в составе архитектуры YOLOR не является оптимальным. Среди всех рассмотренных методов объединения признаков самым производительным является метод TPN, однако анализ работ по внедрению TPN в ResNeXt показывает, что данная архитектура имеет большее повышение точности за счет снижения скорости работы сети [3]. В то же время BiFPN по сравнению с PAN повышает точность на 1.5 AP и скорость модели на 12% [8]. Именно поэтому данная архитектура была выбрана для внедрения в YOLOR взамен PAN. Архитектура BiFPN состоит из чередующихся блоков, структура которых представлена на рисунке 2.3.

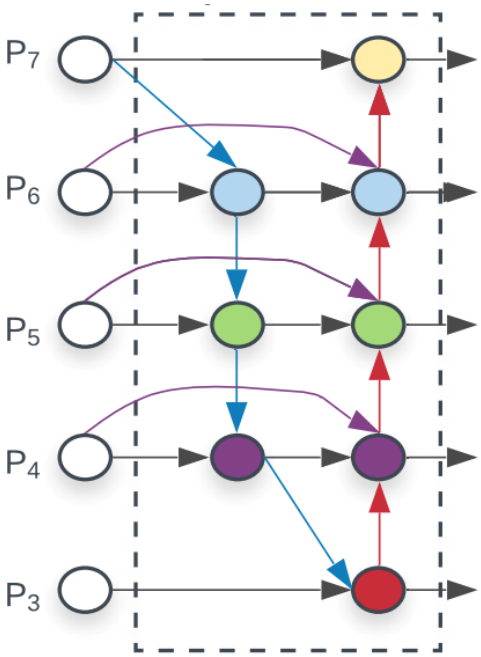

Рисунок 2.3 – Структурная модель блока BiFPN

Данная структура подобно PAN является двунаправленной, однако из нее удалены узлы, имеющие один вход и, следовательно, не участвующие в объединении признаков. Также, добавление межэтапного соединения между входным и выходным слоями, находящимися на одном уровне, позволяет без лишних затрат на вычисление объединять больше признаков. Чередование таких блоков, изображенных на рисунке 2.5 позволяет получить большое количество объединенных признаков высокого уровня [8].

Еще одним изменением шеи детектора является отказ от использования SPP перед этапом обнаружения. Использование SPP изначально предназначалось для устранения ограничения фиксированного размера сети из за использования полностью связанных слоев [6]. В YOLOv4 SPP используется для конкатенации результатов операции подвыборки и увеличение рецептивного поля [1]. В измененной архитектуре в этом нет необходимости, поскольку более глубокая сеть извлечения признаков достаточно увеличивает рецептивное поле. Кроме того, в последних исследования в области сверточных нейронных сетей возрастает тенденция сокращения использования операций подвыборки там, где в этом нет сильной необходимости, поскольку она приводит к потере пространственной информации.

Архитектура головки детектора не изменяется и остается аналогична используемой в YOLOR и YOLOv4 [1].

Предлагаемая измененная модель YOLOR представлена на рисунке 2.4.

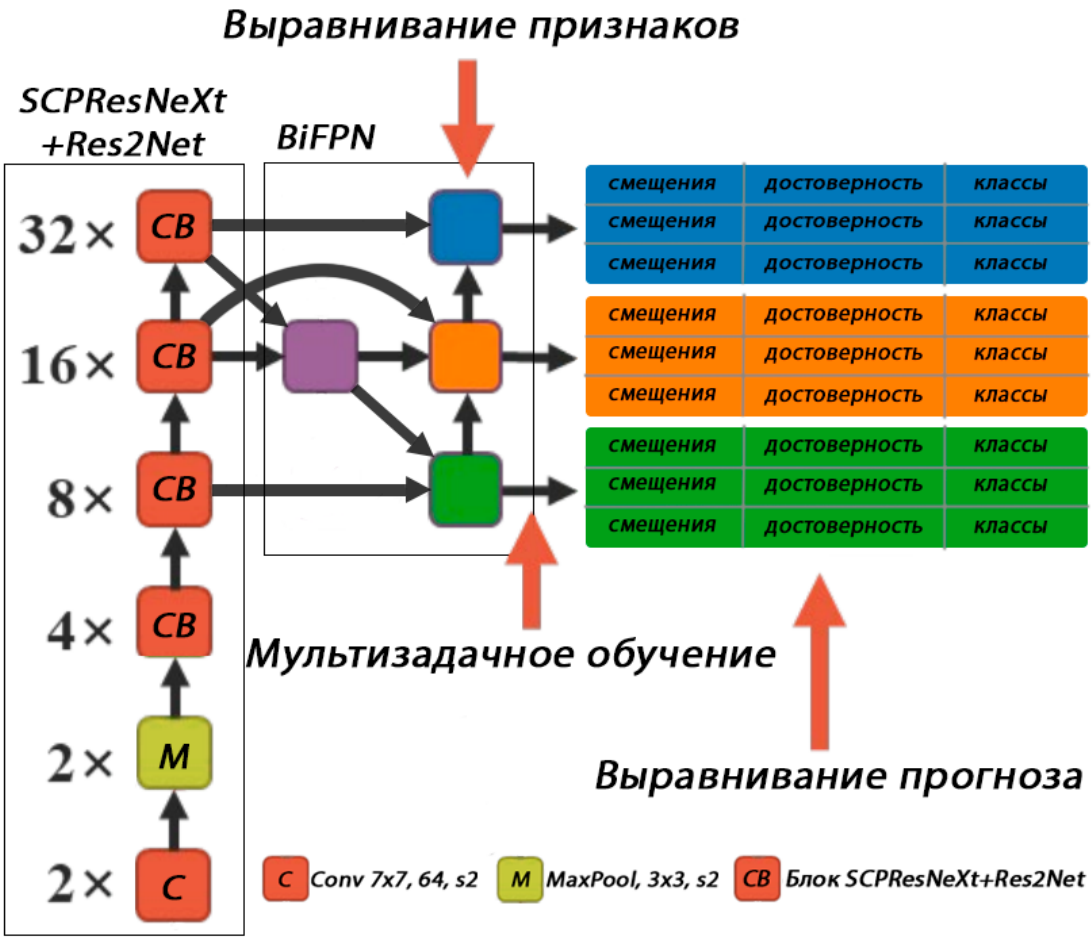

Рисунок 2.4 – Архитектура измененной модели YOLOR

Таким образом было предложен модифицированный YOLOR, содержащий экстрактор признаков CSPResNeXt+Res2Net, который имеет более глубокую сетевую архитектуру при небольшом увеличении количества параметров. Использование данного экстрактора признаков позволит повысить точность модели с незначительной потерей в скорости. В качестве метода уточнения карт признаков вместо PAN был взят BiFPN, который способен увеличить скорость и точность обнаружения. Также было принято решение убрать объединение пространственных пирамид перед детектором для сохранения пространственной информации.

ЗАКЛЮЧЕНИЕ

В ходе исследования алгоритма YOLOR, предназначенного для обнаружения объектов в режиме реального времени, были проанализированы и оценены все его структурные части и произведены следующие изменения:

- экстрактор признаков CSPDarknet-53 был заменен на более глубокую сеть CSPResNet-101, объединяющую подходы ResNeXt и Res2Net, которые создают богатые карты признаков и значительно повышают отзыв модели за счет гибкого масштабирования карт признаков;
- была полностью переработана шея детектора YOLOR, а именно заменен метод уточнения карт признаков с PAN на BiFPN, который показывает значительный прирост к скорости и точности по сравнению с другими сетями пирамид признаков;
- этап SPP, применяемый в YOLOR для увеличения рецептивного поля был удален из модели, поскольку предложенная архитектура экстрактора за счет большей глубины достаточно увеличивает рецептивное поле, кроме того исключение данного этапа из модели приводит к сохранению пространственной информации, которая теряется при выполнении подвыборки.

Подбор оптимальных параметров модели, таких как мощность и масштаб, характеризующих количество ветвей сверток ResNext и количество групп признаков в блоке Res2Net, а также число последовательных блоков BiFPN в сети уточнения признаков будет производится в дальнейших исследованиях в ходе обучения модели.

Таким образом, сконструированная архитектура детектора способна значительно повысить точность модели, сохраняя скорость работы, требуемую для приложений в реальном времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Alexey Bochkovskiy. YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/2004.10934 (дата обращения 01.12.2021).

2. Can Zhang. PAN: Towards Fast Action Recognition via Learning Persistence of Appearance, 2020 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/2008.03462 (дата обращения 10.12.2021).

3. Cédric Picron. Trident Pyramid Networks: The importance of processing at the feature pyramid level for better object detection, 2021 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/2110.04004 (дата обращения 10.12.2021).

4. Chien-Yao Wang. CSPNet: A New Backbone that can Enhance Learning Capability of CNN, 2019 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1911.11929 (дата обращения 01.12.2021).

5. Chien-Yao Wang. You Only Learn One Representation: Unified Network for Multiple Tasks, 2021 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/2105.04206 (дата обращения 01.12.2021).

6. Kaiming He. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, 2014 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1512.03385 (дата обращения 10.12.2021).

7. Kaiming He. Deep Residual Learning for Image Recognition, 2015 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1512.03385 (дата обращения 10.12.2021).

8. Mingxing Tan. EfficientDet: Scalable and Efficient Object Detection, 2019 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1911.09070 (дата обращения 10.12.2021).

9. Saining Xie. Aggregated Residual Transformations for Deep Neural Networks, 2016 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1611.05431 (дата обращения 10.12.2021).

10. Shang-Hua Gao. Res2Net: A New Multi-scale Backbone Architecture, 2019 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1904.01169 (дата обращения 10.12.2021).

11. Tsung-Yi Lin. Feature Pyramid Networks for Object Detection, 201 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1612.03144 (дата обращения 10.12.2021).

12. Yunpeng Chen. Dual Path Networks, 2017 // arXiv [Электронный ресурс]: открытый архив научных статей. URL: https://arxiv.org/abs/1707.01629 (дата обращения 10.12.2021).