



**Министерство науки и высшего образования  
Российской Федерации Федеральное государственное  
бюджетное образовательное учреждение высшего  
образования «Московский государственный  
технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

**ФАКУЛЬТЕТ** Информатика, системы управления и искусственный интеллект

**КАФЕДРА** Системы обработки информации и управления

**Домашнее задание  
По курсу  
«Методы машинного обучения в АСОИУ»**

Выполнил:  
Студент группы ИУ5-22М  
Кириллов Д.С.  
06.05.2024

Проверил:  
**Гапанюк Ю.Е.**

Москва 2024 г.

---

## Оглавление

Оглавление .....	1
Задание .....	2
Постановка задачи.....	4
Теоретическая часть.....	5
Практическая часть .....	7
Выводы .....	15
Список источников .....	16

## Задание

Домашнее задание по дисциплине направлено на анализ современных методов машинного обучения и их применение для решения практических задач. Домашнее задание включает три основных этапа:

1. выбор задачи;
2. теоретический этап;
3. практический этап.

Этап выбора задачи предполагает анализ ресурса `paperswithcode`. Данный ресурс включает описание нескольких тысяч современных задач в области машинного обучения. Каждое описание задачи содержит ссылки на наиболее современные и актуальные научные статьи, предназначенные для решения задачи (список статей регулярно обновляется авторами ресурса). Каждое описание статьи содержит ссылку на репозиторий с открытым исходным кодом, реализующим представленные в статье эксперименты. На этапе выбора задачи обучающийся выбирает одну из задач машинного обучения, описание которой содержит ссылки на статьи и репозитории с исходным кодом.

Теоретический этап включает проработку как минимум двух статей, относящихся к выбранной задаче. Результаты проработки обучающийся излагает в теоретической части отчета по домашнему заданию, которая может включать:

- описание общих подходов к решению задачи;
- конкретные топологии нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения, предназначенных для решения задачи;
- математическое описание, алгоритмы функционирования, особенности обучения используемых для решения задачи нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения;
- описание наборов данных, используемых для обучения моделей;

- оценка качества решения задачи, описание метрик качества и их значений;
- предложения обучающегося по улучшению качества решения задачи.

Практический этап включает повторение экспериментов авторов статей на основе представленных авторами репозитория с исходным кодом и возможное улучшение обучающимися полученных результатов. Результаты проработки обучающийся излагает в практической части отчета по домашнему заданию, которая может включать:

- исходные коды программ, представленные авторами статей, результаты документирования программ обучающимися с использованием диаграмм UML, путем визуализации топологий нейронных сетей и другими способами;
- результаты выполнения программ, вычисление значений для описанных в статьях метрик качества, выводы обучающегося о воспроизводимости экспериментов авторов статей и соответствии практических экспериментов теоретическим материалам статей;
- предложения обучающегося по возможным улучшениям решения задачи, результаты практических экспериментов (исходные коды, документация) по возможному улучшению решения задачи.

Отчет по домашнему заданию должен содержать:

- Титульный лист.
- Постановку выбранной задачи машинного обучения, соответствующую этапу выбора задачи.
- Теоретическую часть отчета.
- Практическую часть отчета.
- Выводы обучающегося по результатам выполнения теоретической и практической частей.
- Список использованных источников.

## **Постановка задачи**

По данным Всемирной организации здравоохранения (ВОЗ), для сдерживания распространения вируса рекомендуется использовать медицинские маски в сочетании с другими мерами по охране здоровья. В этом контексте можно создать видео-детектор, который будет определять правильно и неправильно надетые медицинские маски в режиме реального времени. Это задача сегментации на видеопотоке.

## Теоретическая часть

В соответствии с темой была подобрана работа [1].

В данной работе описывается обучение нескольких популярных моделей для сегментации объектов на видео и производится сравнение их производительности, fps, качества IoU, mAP и стоимости в ресурсах: вычислительной, GFLOPS, по памяти, Гб

Сверточные нейронные сети (CNN) широко используются для решения задач обнаружения объектов благодаря их повышенной точности. Тем не менее, эффективность моделей обнаружения на основе CNN неоднозначна, если учитывать скорость обнаружения. Насколько нам известно, в соответствующей литературе не проводилось достаточной оценки доступных методов с точки зрения соотношения скорости и точности. В данной работе рассматриваются наиболее фундаментальные модели обнаружения объектов на основе общих Объекты в контекстном наборе данных (COCO) с учетом этого компромисса, их потребление памяти, а также вычислительные затраты и затраты на хранение. Затем мы выбираем высокоэффективную модель под названием YOLOv5 для обучения на актуальном и неисследованном наборе данных о лицах людей в медицинских масках, наборе данных о лицах, носящих маски надлежащим образом (PWMFD), и анализируем преимущества конкретных методов оптимизации для обнаружения медицинских масок в режиме реального времени: обучение переносу, увеличение объема данных и Механизм сжатия и возбуждения внимания. Используя наши результаты в контексте пандемии COVID-19, мы предлагаем оптимизированную модель, основанную на YOLOv5s, использующую перенос обучение распознаванию правильно и неправильно надетых медицинских масок, которое более чем в два раза превзошло по скорости (69 кадров в секунду) современную модель SE-YOLOv3 из набора данных PWMFD при сохранении того же среднего уровня Средняя точность (67%).

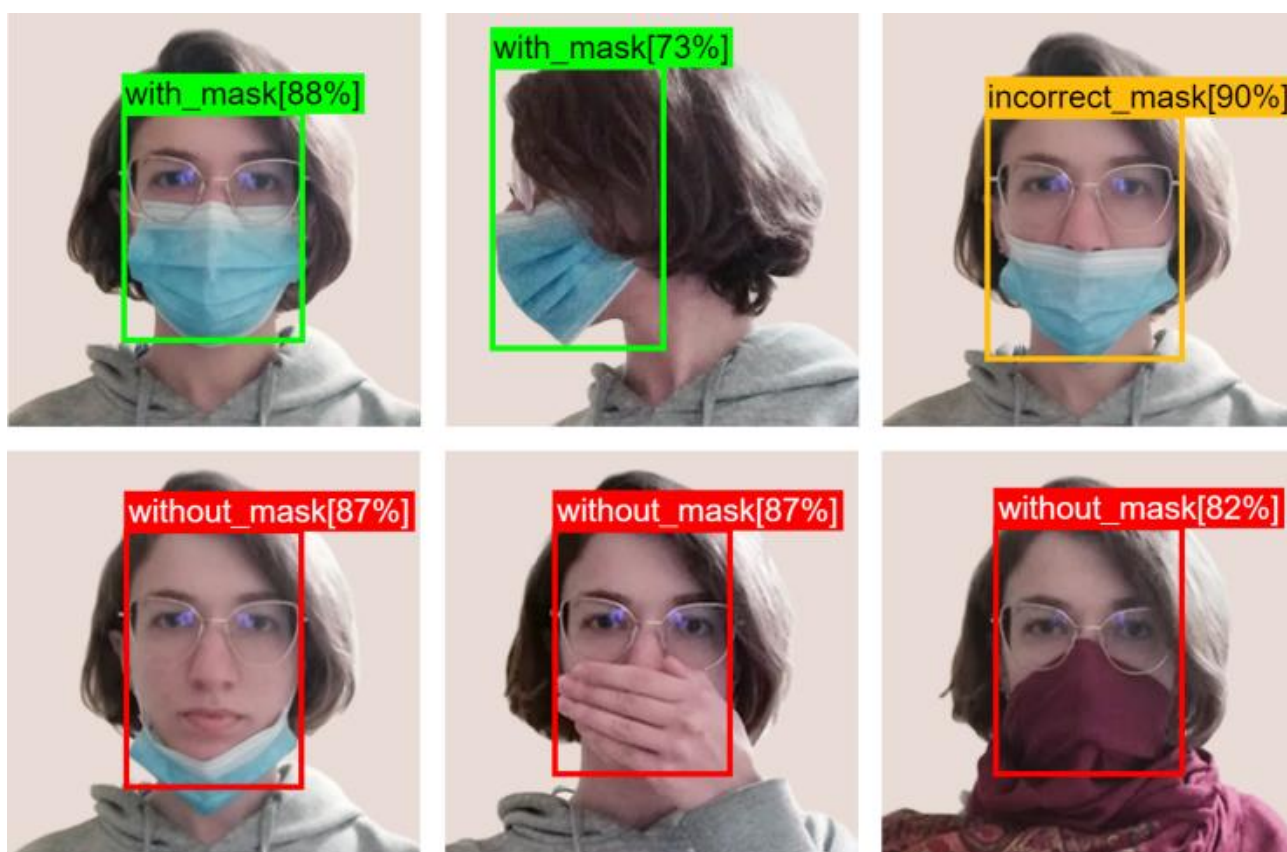


Рис.1. Иллюстрация выполнения задачи распознавания медицинской маски от авторов статьи [1].

## Практическая часть

Авторы приводят 3 `ipynb`-документа для запуска в Colab и 60 таблиц: результаты измерений производительности, качества и стоимости для 15 моделей на двух датасетах и двух видеокартах.

Один из `ipynb`-ов, отвечающий за анализ результатов, был запущен в колабе.

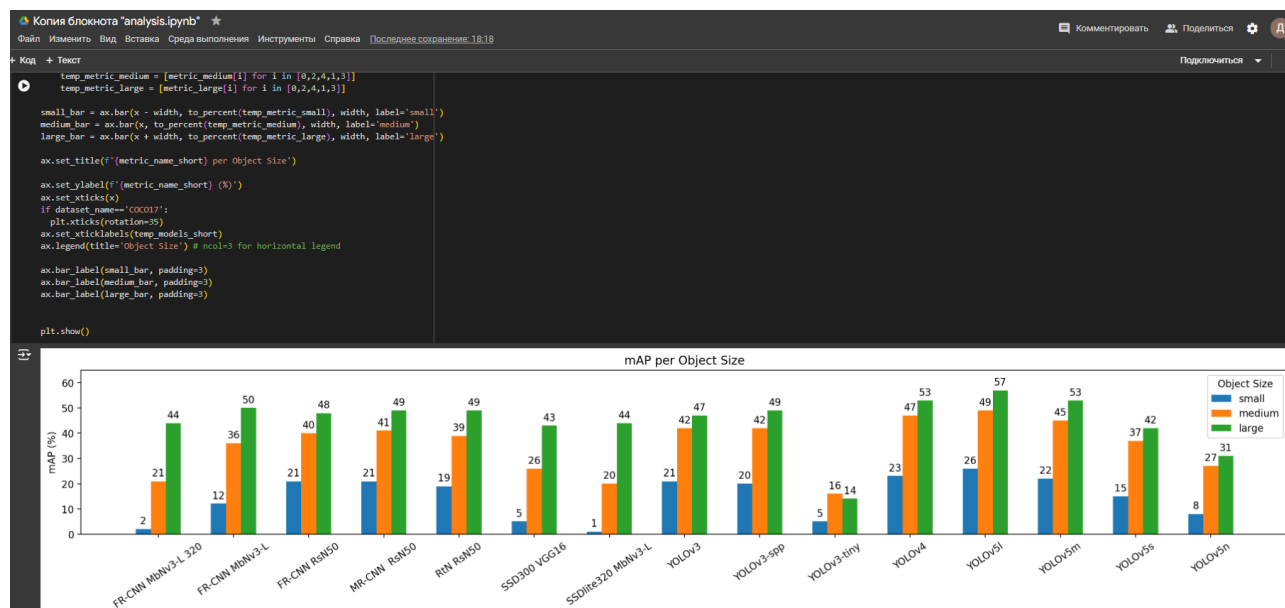


Рис. 2. Запуск `analyze.ipynb` от авторов [1] на собранном ими наборе данных.

Скрипт соединяет 30 таблиц с результатами в одну. Итоговый датасет содержит 380 строк. Ниже приведена таблица 1 с фрагментом собранного датасета. В таблице 2 представлены обработанные данные из таблицы 1, по которым строятся рисунки 3-12.



Таблица 1. Фрагмент собранного датасета.

<b>model</b>	<b>model_img_ size</b>	<b>model_par ams</b>	<b>model_gfl ops</b>	<b>fps</b>	<b>max_m em</b>	<b>iou_ty pe</b>	<b>metric</b>	<b>iou</b>	<b>area</b>	<b>max_d ets</b>	<b>score</b>	<b>gpu</b>
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_precis ion	0.50:0. 95	all	100	0,2276 53	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_precis ion	0,5	all	100	0,3797 78	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_precis ion	0,75	all	100	0,2319 55	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_precis ion	0.50:0. 95	small	100	0,0258 6	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_precis ion	0.50:0. 95	mediu m	100	0,2179 85	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_precis ion	0.50:0. 95	large	100	0,4428 6	gtx9 60

model	model_img_ size	model_par ams	model_gfl ops	fps	max_m em	iou_ty pe	metric	iou	area	max_d ets	score	gpu
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_recall	0.50:0. 95	all	1	0,2154 57	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_recall	0.50:0. 95	all	10	0,2901 68	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_recall	0.50:0. 95	all	100	0,2937 02	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_recall	0.50:0. 95	small	100	0,0372 78	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_recall	0.50:0. 95	mediu m	100	0,2956 47	gtx9 60
Faster R-CNN MobileNetV3- Large 320 FPN	320	19,4	1,4	41,964 49	195,669 4	bbox	avg_recall	0.50:0. 95	large	100	0,569	gtx9 60

model	model_img_ size	model_par ams	model_gfl ops	fps	max_m em	iou_ty pe	metric	iou	area	max_d ets	score	gpu
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_precis ion	0.50:0. 95	all	100	0,3278 35	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_precis ion	0,5	all	100	0,5252 6	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_precis ion	0,75	all	100	0,3433 94	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_precis ion	0.50:0. 95	small	100	0,1273 46	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_precis ion	0.50:0. 95	mediu m	100	0,3630 91	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_precis ion	0.50:0. 95	large	100	0,5018 98	gtx9 60

model	model_img_ size	model_par ams	model_gfl ops	fps	max_m em	iou_ty pe	metric	iou	area	max_d ets	score	gpu
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_recall	0.50:0. 95	all	1	0,2866 23	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_recall	0.50:0. 95	all	10	0,4264 18	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_recall	0.50:0. 95	all	100	0,4443 6	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_recall	0.50:0. 95	small	100	0,1960 22	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_recall	0.50:0. 95	mediu m	100	0,4989 28	gtx9 60
Faster R-CNN MobileNetV3- Large FPN	800	19,4	9	12,241 78	304,757 3	bbox	avg_recall	0.50:0. 95	large	100	0,6482 51	gtx9 60

model	model_img_ size	model_par ams	model_gfl ops	fps	max_m em	iou_ty pe	metric	iou	area	max_d ets	score	gpu
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,5489 78	593,809 6	bbox	avg_precis ion	0.50:0. 95	all	100	0,3695 06	gtx9 60
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,5489 78	593,809 6	bbox	avg_precis ion	0,5	all	100	0,5857 94	gtx9 60
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,5489 78	593,809 6	bbox	avg_precis ion	0,75	all	100	0,3964 19	gtx9 60
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,5489 78	593,809 6	bbox	avg_precis ion	0.50:0. 95	small	100	0,2124 44	gtx9 60
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,5489 78	593,809 6	bbox	avg_precis ion	0.50:0. 95	mediu m	100	0,4032 69	gtx9 60
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,5489 78	593,809 6	bbox	avg_precis ion	0.50:0. 95	large	100	0,4818 13	gtx9 60

model	model_img_size	model_params	model_gflops	fps	max_mem	iou_type	metric	iou	area	max_det	score	gpu
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,5489 78	593,809 6	bbox	avg_recall	0.50:0. 95	all	1	0,3074 62	gtx9 60

Таблица 2. Сгруппированная выборка из таблицы 1 для построения графиков производительности на видеоадаптере gtx960.

model	model_img_size	model_params	model_gflops	fps	max_mem	map	map_50	map_75	mar
Faster R-CNN MobileNetV3-Large 320 FPN	320	19,4	1,4	41,96	195,67	0,23	0,38	0,23	0,29
Faster R-CNN MobileNetV3-Large FPN	800	19,4	9	12,24	304,76	0,33	0,53	0,34	0,44
Faster R-CNN ResNet-50 FPN	800	41,8	240,4	3,55	593,81	0,37	0,59	0,4	0,51
Mask R-CNN ResNet-50 FPN	800	44,4	259,9	3,29	603,65	0,38	0,59	0,41	0,52
RetinaNet ResNet-50 FPN	800	34	303,5	3,2	428,73	0,36	0,56	0,38	0,54
SSD300 VGG16	300	35,6	69,8	14,84	209,81	0,25	0,42	0,26	0,36
SSDlite320 MobileNetV3-Large	320	3,4	1,2	26,1	33,27	0,21	0,34	0,22	0,33
YOLOv3	640	61,9	156,3	11,09	300,77	0,37	0,52	0,4	0,44
YOLOv3-spp	640	63	157,1	11,92	306,23	0,38	0,54	0,41	0,44
YOLOv3-tiny	640	8,8	13,3	69,37	91,42	0,12	0,22	0,12	0,15
YOLOv4	608	64,4	128,9	6,88	924,79	0,41	0,57	0,47	0,44
YOLOv5l	640	46,5	109,2	14,54	403,34	0,44	0,59	0,48	0,5
YOLOv5m	640	21,2	49	25,92	190,92	0,4	0,56	0,44	0,46

<b>model</b>	<b>model_img_s ize</b>	<b>model_para ms</b>	<b>model_gflo ps</b>	<b>fps</b>	<b>max_me m</b>	<b>map</b>	<b>map_ 50</b>	<b>map_ 75</b>	<b>mar</b>
YOLOv5s	640	7,2	16,5	50,35	78,19	0,32	0,46	0,35	0,38
YOLOv5n	640	1,9	4,5	88,33	49,49	0,23	0,36	0,25	0,28

В ноутбуке генерируются следующие графики, лишь один из которых авторы приводят в публикации (рис. 3-12).

Гистограмма производительности моделей на различных GPU, в данном случае на GPU GTX960.

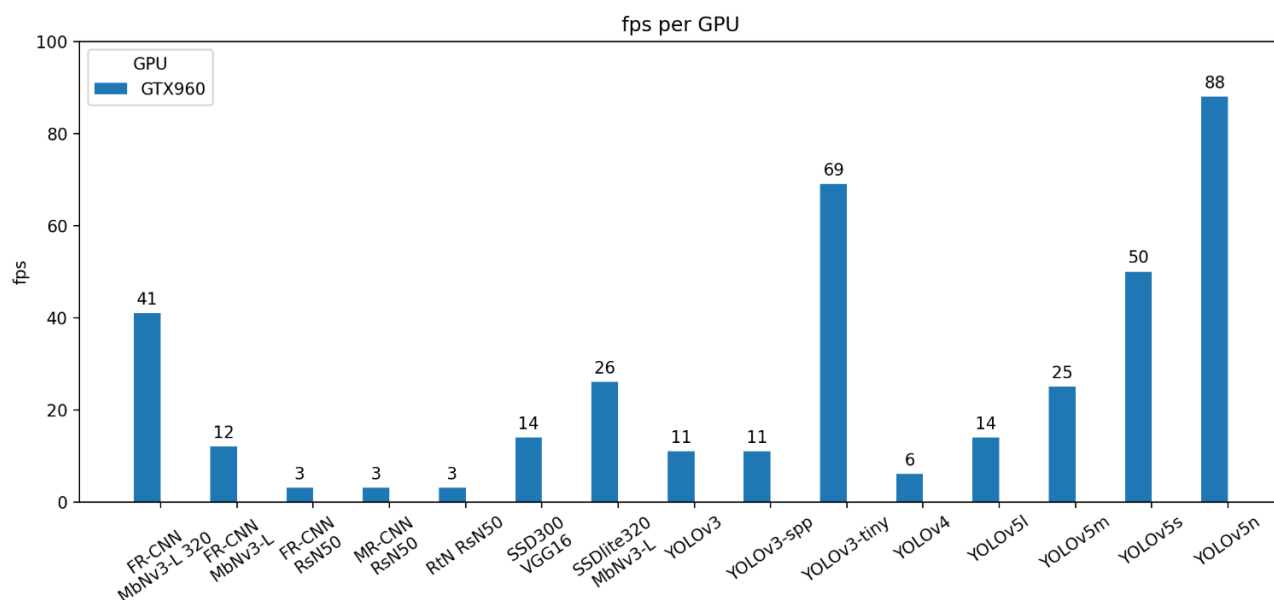


Рис.3. Plot fps per GPU.

Диаграмма разброса точности mAP и производительность различных популярных моделей.



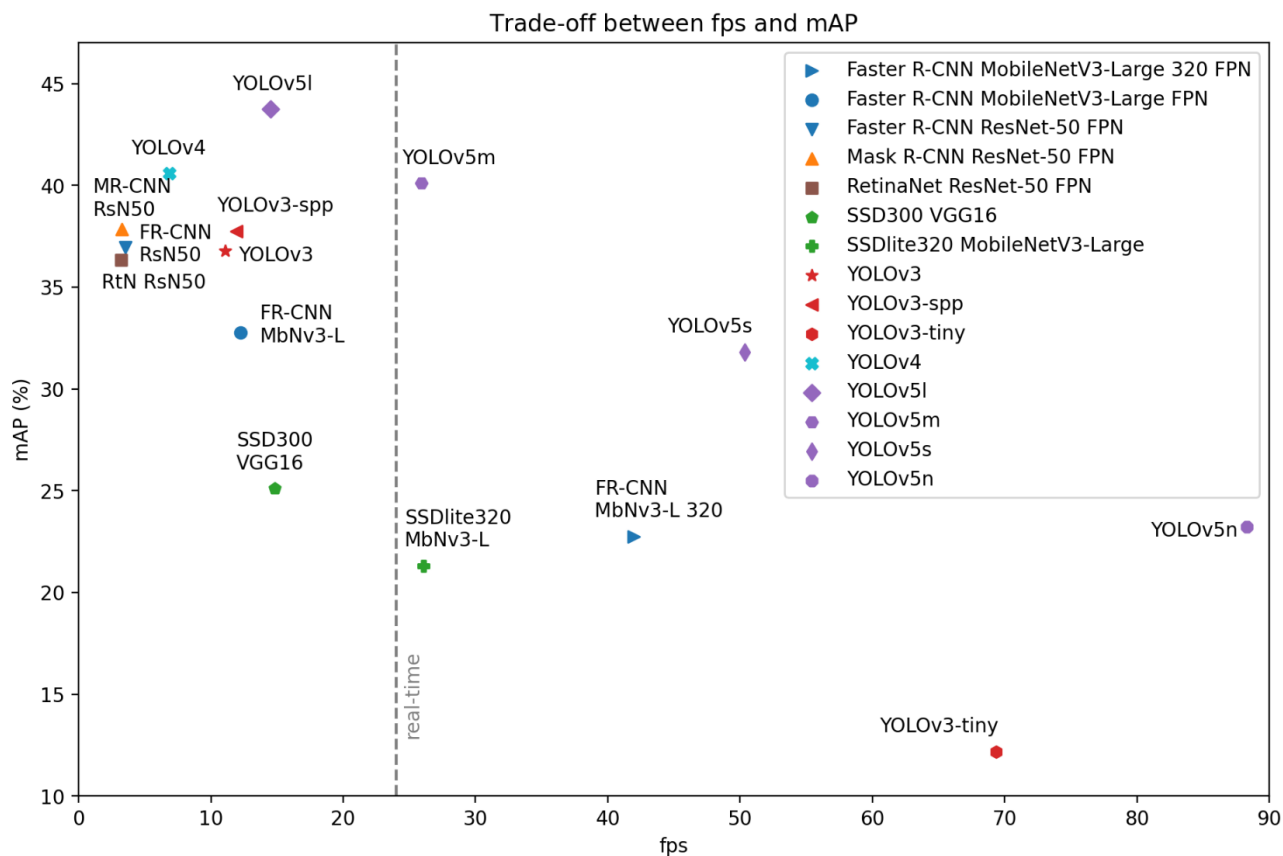


Рис.4. Plot mAP / fps or mAR / fps

Гистограмма среднеклассовой точности в зависимости от размеров объектов на изображениях для различных популярных моделей.

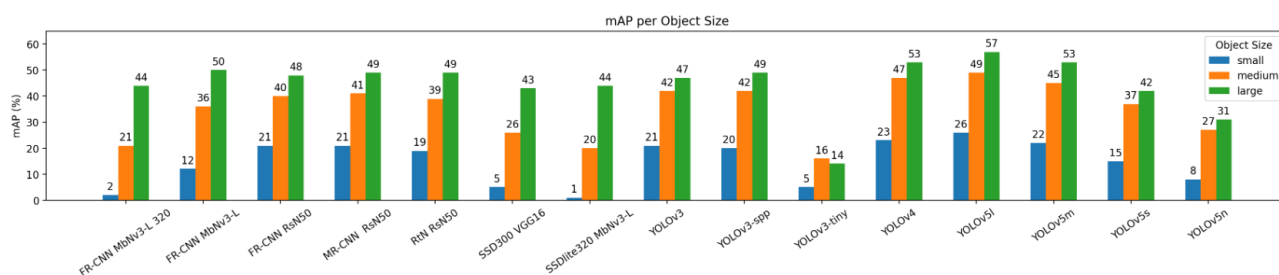


Рис.5. Plot mAP or mAR (small,medium,large) per model (change parameters)

Гистограмма среднеклассовой точности в зависимости от порога уверенности по IoU для различных популярных моделей.

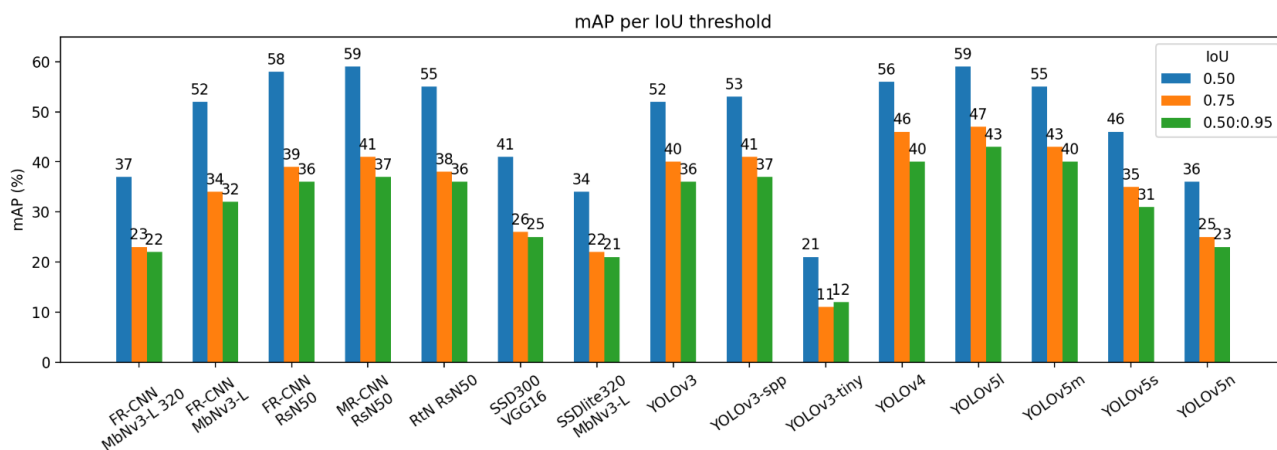


Рис.6. Plot mAP (IoU = [0.50,0.75,0.50:0.95]) per model.

Гистограмма среднеклассовой точности в зависимости от числа детекций для различных популярных моделей.

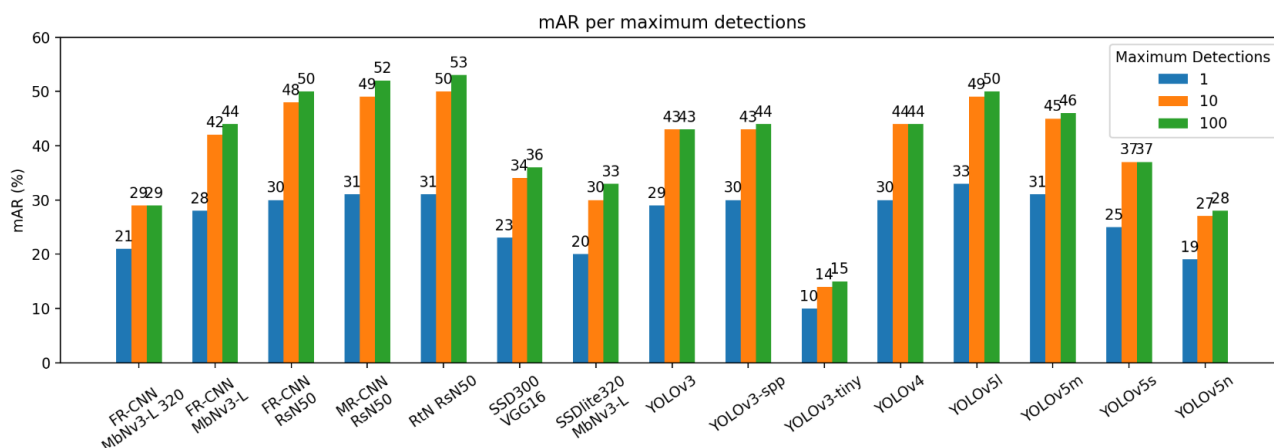


Рис.7. Plot mAR (max det = [1,10,100]) per model.

Диаграмма разброса среднеклассовой точности и вычислительной стоимости для различных популярных моделей.

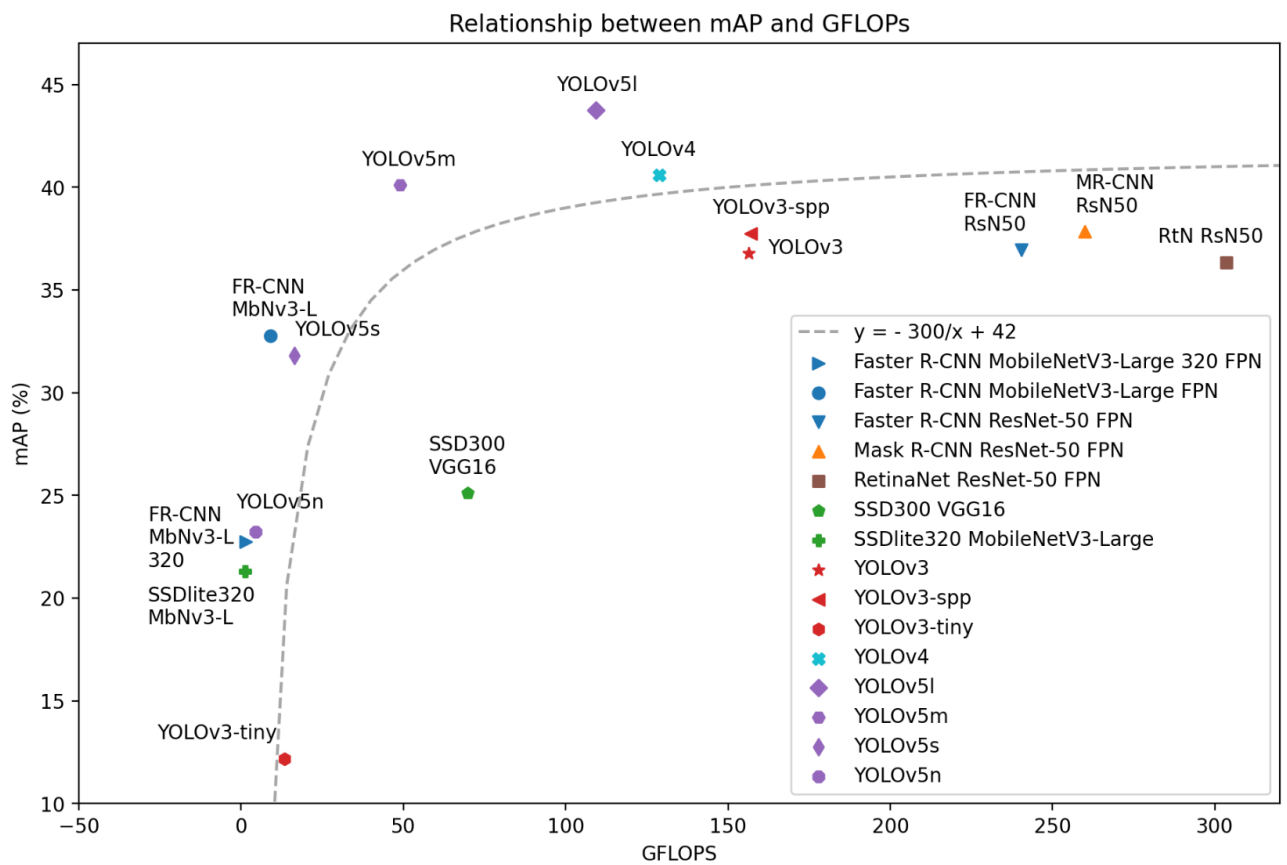


Рис.8. Plot model mAP / GFLOPs.

Диаграмма разброса производительности и вычислительной стоимости для различных популярных моделей.

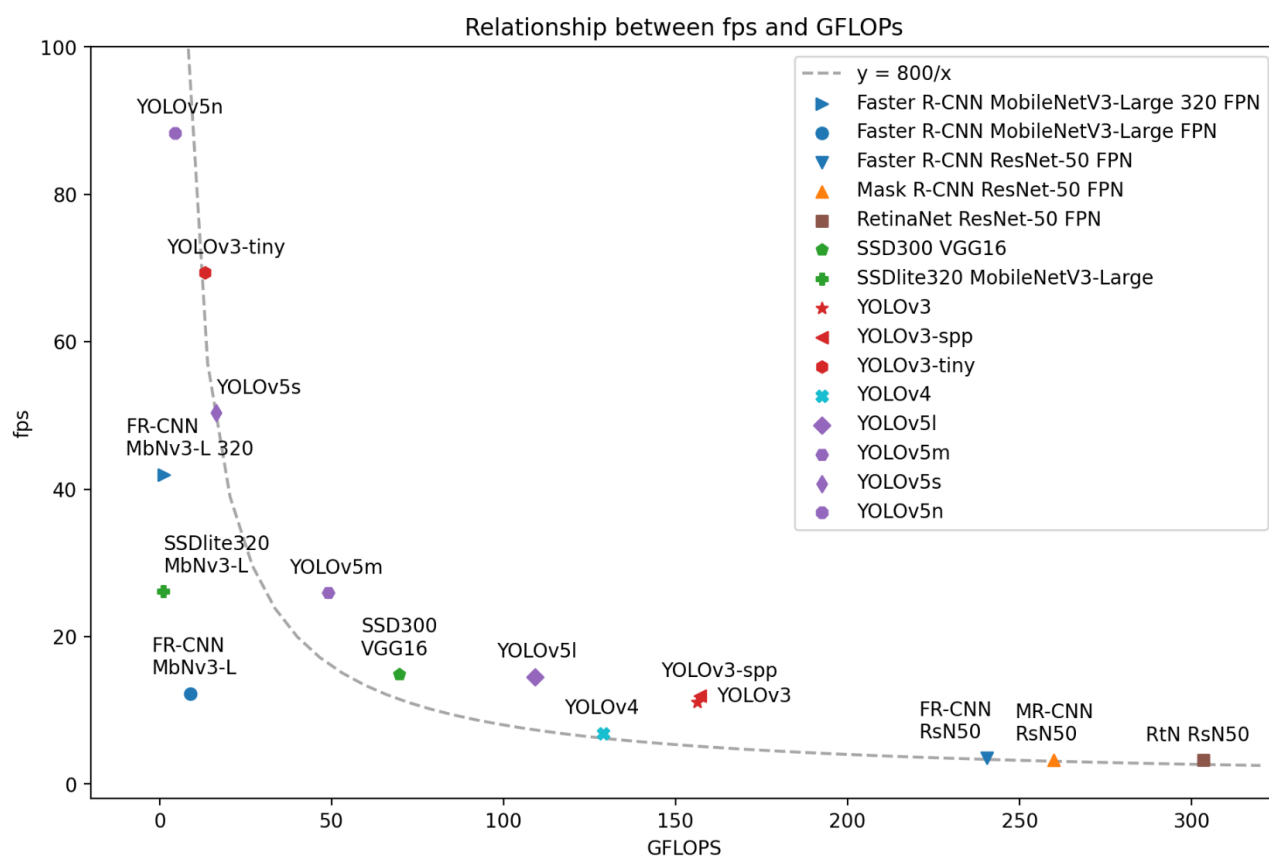


Рис.9. Plot model fps / GFLOPs.

Диаграмма разброса числа параметров и размера изображений с тепловой картой вычислительной стоимости для различных популярных моделей. Такая диаграмма демонстрирует вычислительную стоимость.

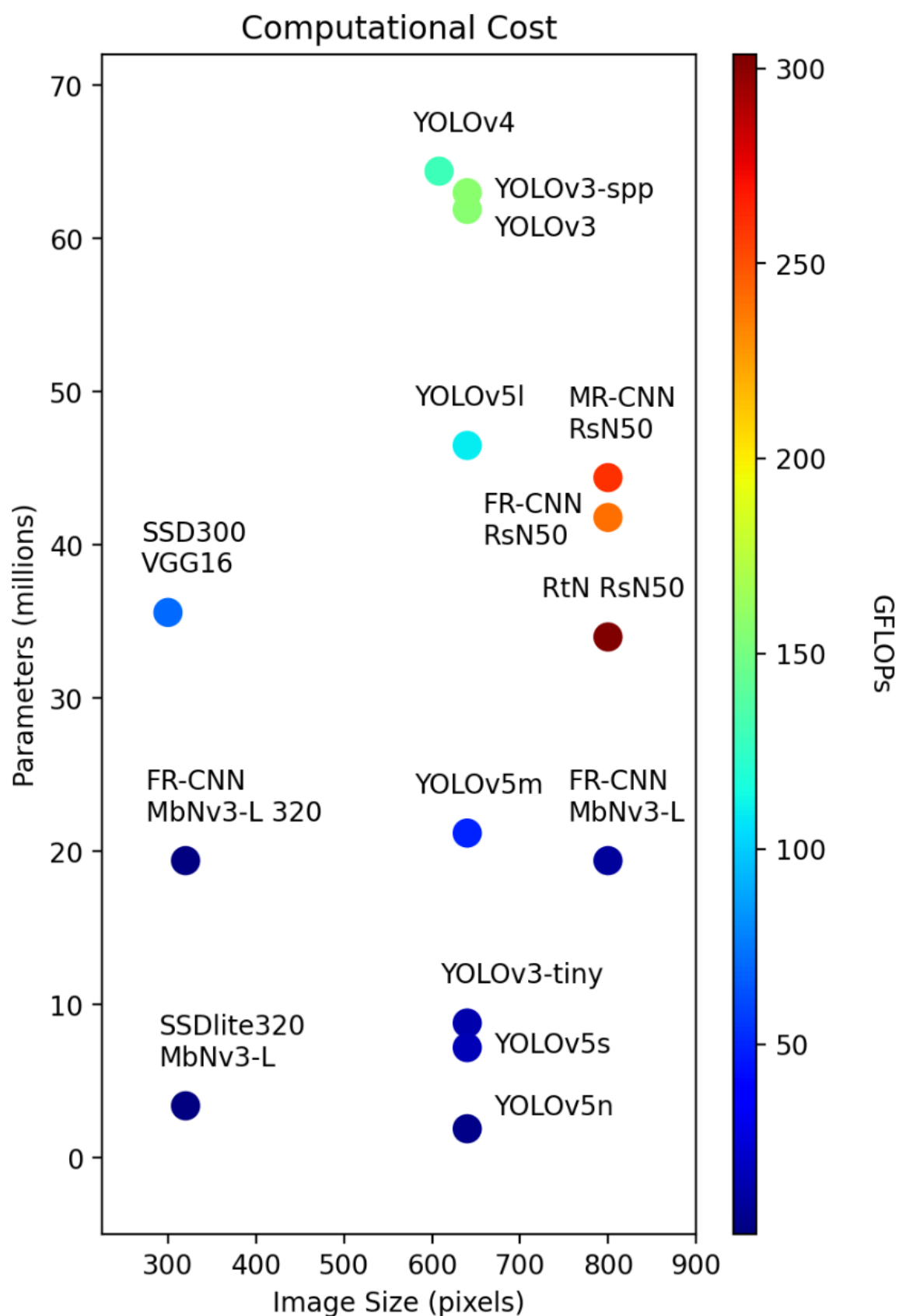


Рис.10. Plot GFLOPs.

Диаграмма разброса числа параметров и размера изображений с тепловой картой потребляемого объема памяти для различных популярных моделей. Такая

диаграмма демонстрирует стоимость использования моделей по памяти.

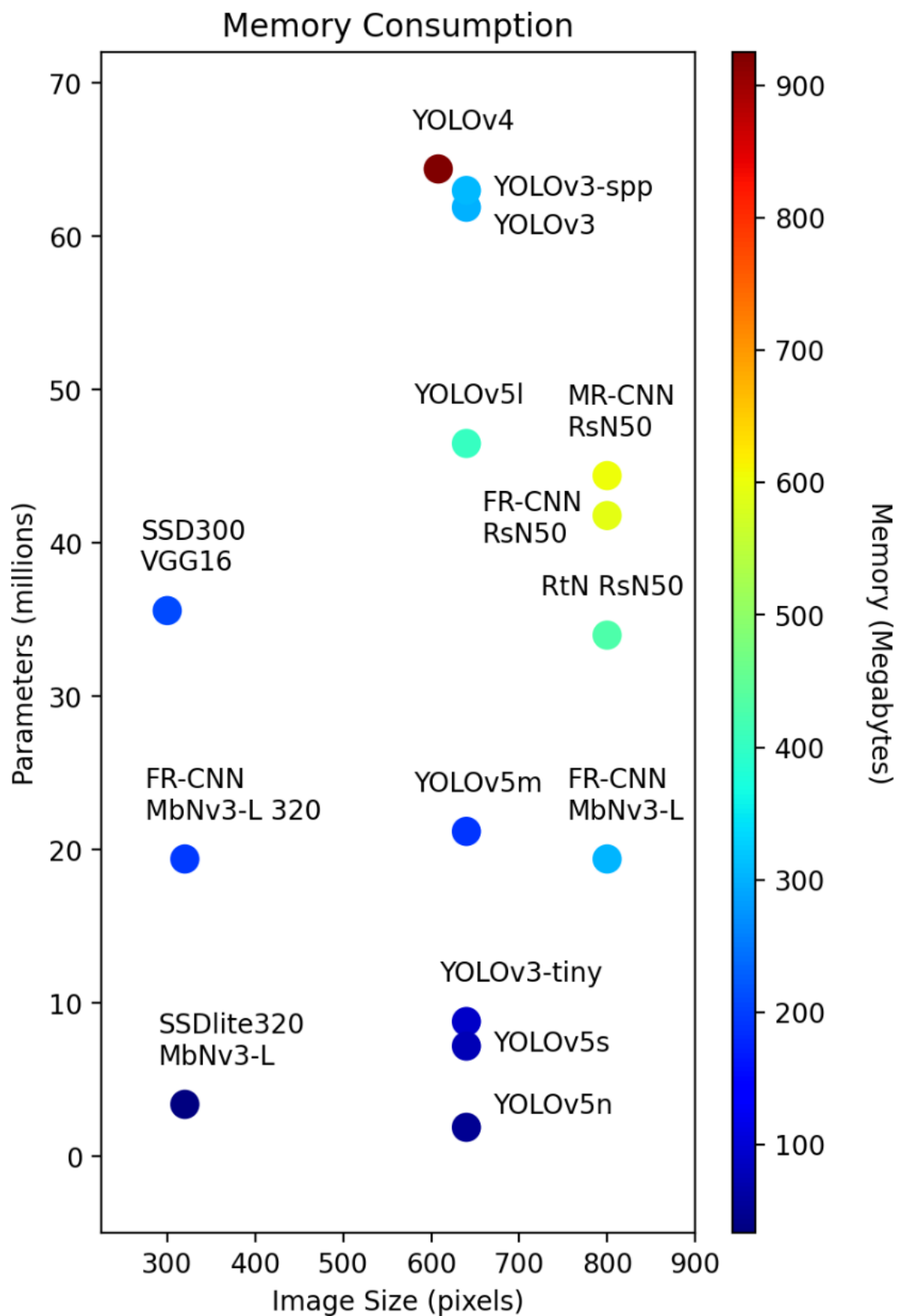


Рис.11. Plot Memory usage.

Диаграмма разброса числа параметров и размера модели на диске для

различных популярных моделей.

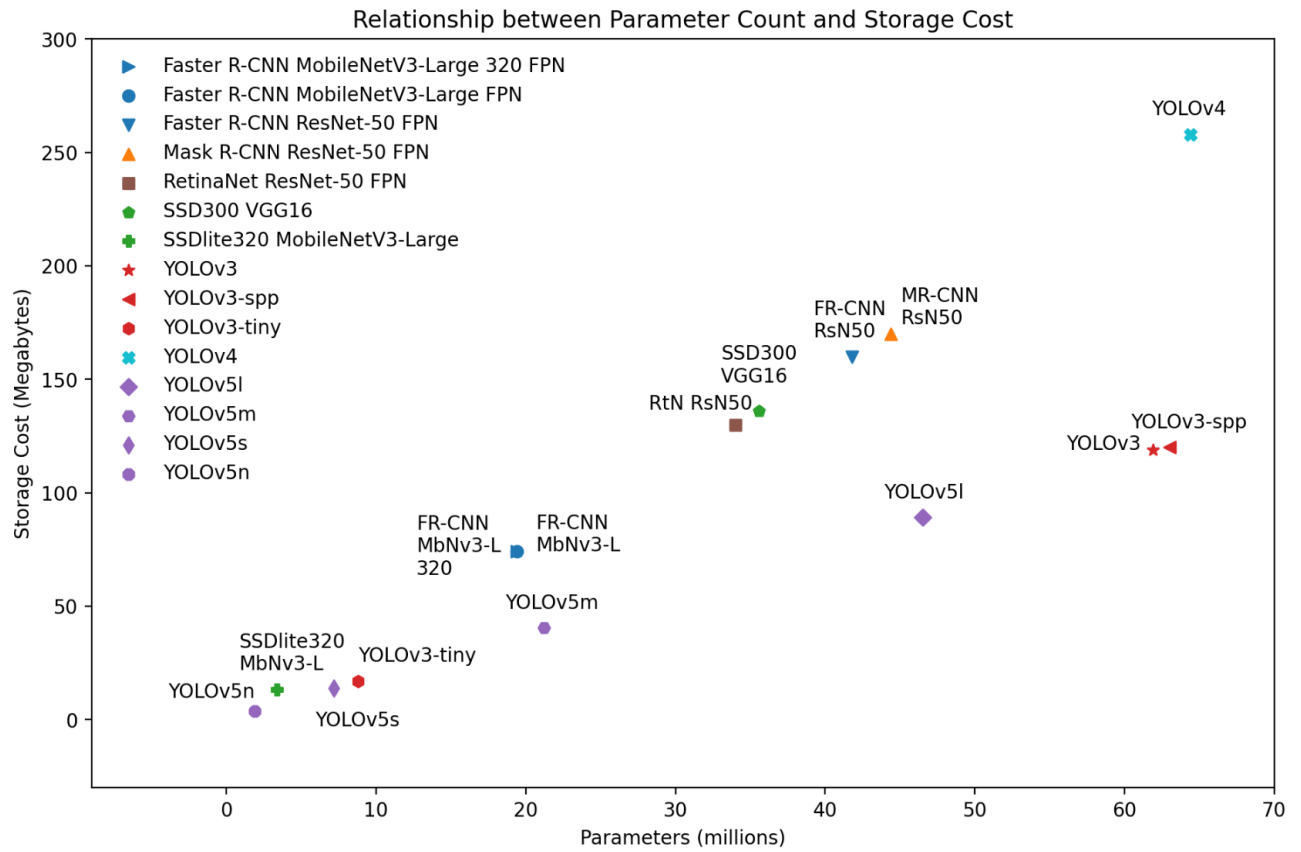


Рис.12. Plot Storage / Parameters.

## **Выводы**

В ходе выполнения работы ознакомились с методами обучения с подкреплением на основе глубоких Q-сетей с помощью библиотеки Gym.

В работе было произведено ознакомление с научной работой [1] для сегментации изображений в видеопотоке. В [1] авторы создали видео-детектор, который определяет правильно и неправильно надетые медицинские маски в режиме реального времени.



## **Список источников**

1. I. C. Gogou and D. A. Koutsomitropoulos, "A Review and Implementation of Object Detection Models and Optimizations for Real-time Medical Mask Detection during the COVID-19 Pandemic," 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Biarritz, France, 2022, pp. 1-6, doi: 10.1109/INISTA55318.2022.9894232. <https://paperswithcode.com/paper/a-review-and-implementation-of-object>