Московский государственный технический университет им. Н.Э.

Баумана Кафедра «Системы обработки информации и управления»

Домашнее задание

по дисциплине

«Методы машинного обучения»

Выполнил:

студент группы ИУ5И-22М

Чжан Аньци

Москва — 2022 г.

Содержание

1. этап выбора задачи

Задача выбора - Классификация изображений.

Технологии компьютерного зрения очень распространены. Они применяются для распознавания лиц, пешеходов, объектов, для медицинского анализа, навигации автономных автомобилей и во многих других сферах. В связи с ростом вычислительных мощностей и появлением больших баз изображений стало возможным обучать глубокие нейронные сети — нейронные сети с большим числом скрытых слоёв. В задаче распознавания образов особого успеха достигли свёрточные нейронные сети (Convolutional Neural Networks), которые каждый год с 2012 года выигрывали соревнование ImageNet Large Scale Visual Classification Challenge (ILSVRC) [Russakovsky et al., 2015].

Одной из базовых задач в машинном зрении является задача классификации изображения — определения категорий объектов, который находится на изображении. В зависимости от конкретной задачи, на изображении может быть аннотирован как один объект, так и несколько.

Для оценки алгоритмов машинного обучения обычно используются аннотированные базы изображений, например, CIFAR-10 [Krizhevsky, 2009], ImageNet [Russakovsky et al., 2015], PASCAL VOC [Everingham

et al., 2010].

Классификация изображений - это фундаментальная задача, которая пытается понять все изображение как единое целое. Цель - классифицировать изображение, присвоив ему определенную метку. Как правило, классификация изображений относится к изображениям, на которых отображается и анализируется только один объект. В отличие от этого, обнаружение объектов включает в себя задачу классификации и локализации и используется для анализа более реалистичных ситуаций, когда на изображении может присутствовать несколько объектов.

Классификацией изображений называется метод обработки изображений, который позволяет различать различные классы объектов на основе различных характеристик, которые каждый из них отражает в информации изображения. В нем используется компьютер для количественного анализа изображения и классификации изображения или каждого элемента изображения или области внутри изображения в одну из нескольких категорий вместо визуальной интерпретации человеком.

Классификация изображений является одной из самых фундаментальных задач в компьютерном зрении и задачей, с которой сравниваются почти все эталонные модели. Начиная с относительно простой задачи mnist, состоящей из 10 категорий, для распознавания

рукописных цифр на изображениях в сером масштабе, до более крупных задач cifar10 с 10 категориями и cifar100 со 100 категориями, а затем и до задачи imagenet, модели классификации изображений развивались шаг за шагом с ростом наборов данных и достигли того уровня, на котором они находятся сегодня. Сейчас, имея набор данных из более чем 10 миллионов изображений и более 20 000 классов, таких как imagenet, компьютеры превзошли человека в классификации изображений.

Классификация изображений - это, как следует из названия, проблема классификации образов, и ее цель - классифицировать различные изображения в различные классы для достижения наименьшей возможной ошибки классификации. В целом, для задач классификации изображений с одной меткой их можно разделить на три основные категории: межвидовая классификация изображений на семантическом уровне, классификация изображений на уровне подклассов и классификация изображений на уровне экземпляра.

Для классификации изображений обычно используются следующие методы: методы индексирования на основе цветовых признаков, методы классификации изображений на основе текстуры, методы классификации изображений на основе формы и методы классификации изображений на основе пространственных отношений.

Задача классификации изображений состоит в том, чтобы установить соответствие между пикселями и семантикой, преодолев "семантический разрыв". Существуют также вопросы перспективы, освещения, масштаба, окклюзии, деформации, фонового беспорядка, внутриклассовой деформации, размытия движения и широкий спектр категорий.

На сайте paperswithcode представлено 2358 статей с кодом, 106 эталонов и 172 набора данных для классификации изображений. Как показано на рисунке 1.

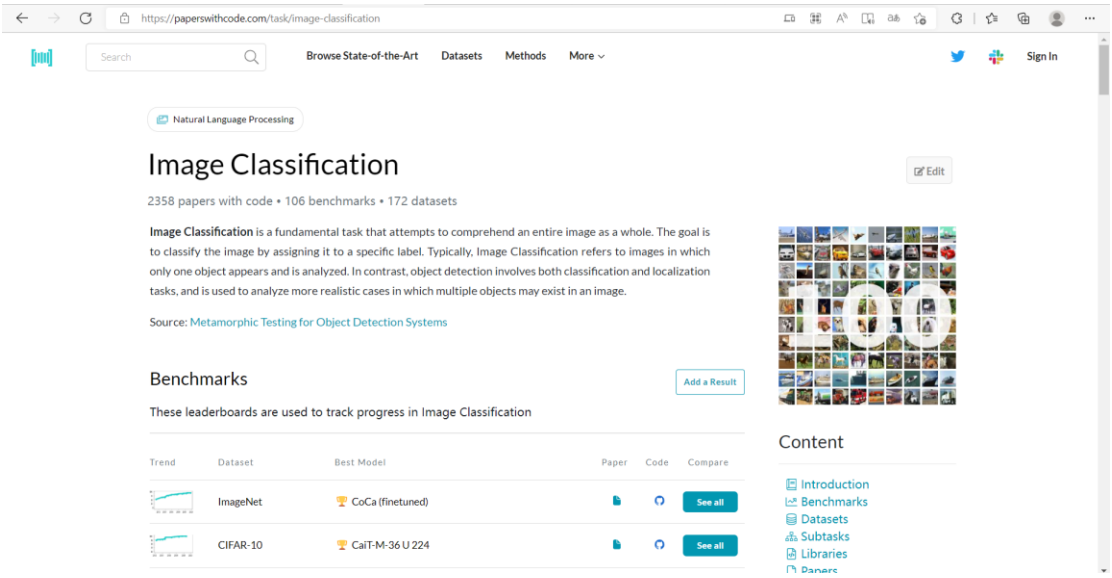

Рис 1 Страница задачи классификации изображений на сайте paperswithcode.com

Похоже, эта тема интересует многих людей, и я в их числе.

2. теоретический этап

Наиболее используемым методом классификации изображений являются сверточные нейронные сети, и две выбранные мной статьи также основаны на сверточных нейронных сетях.

Свёрточная нейронная сеть — нейронная сеть, в которой присутствует слой свёртки (convolutional layer). Обычно в свёрточных нейронных сетях также присутствуют слой субдискретизации (pooling layer) и полносвязный слой (fully connected layer). Свёрточные нейронные сети применяются для оптического распознавания образов [LeCun et al., 1998], классификации изображений [Russakovsky et al., 2015], детектирования предметов [Girshick et al., 2014], семантической сегментации [Long et al., 2015] и других задач [Gu et al., 2017].

Основы современной архитектуры свёрточных нейронных сетей были заложены в одной из первой широко известной свёрточной нейронной сети — LeNet-5 Яна ЛеКуна [LeCun et al., 1998], архитектура которой представлена на рисунке 2.

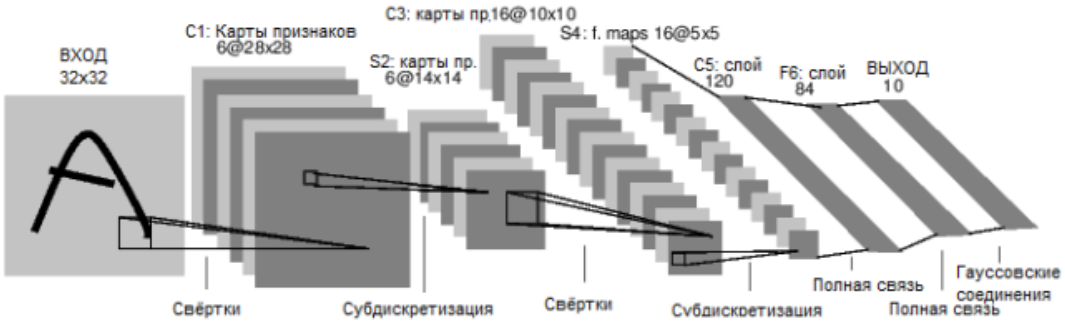

Рис 2 Структура LeNet-5

В свёрточных нейронных сетях слои свёртки и субдискретизации состоят из нескольких «уровней» нейронов, называемых картами признаков (feature maps), или каналами (channels). Каждый нейрон такого слоя соединён с небольшим участком предыдущего слоя, называемым рецептивным полем. В случае изображения, карта признаков является двумерным массивом нейронов, или просто матрицей. Другие измерения могут быть использованы, если на вход принимается другой вид данных, например, аудио данные (одномерный массив) или объёмные данные (трёхмерный массив). В слое свёртки каждой карте признаков соответствует одно ядро свёртки, также называемое фильтром. Каждый нейрон в качестве своего выходного значения осуществляет операцию свёртки или взаимной корреляции со своим рецептивным слоем. Стоит заметить, что эти две операции в контексте обучения свёрточных нейронных сетей взаимозаменяемы, вследствие чего во многих программных реализациях операция “свёртки” на самом деле является операцией взаимной корреляции. Так как ядро свёртки для каждой карты

признаков одно, это позволяет нейронной сети научиться выделять признаки вне зависимости от их расположения во входном изображении и также приводит к значительному уменьшению числа параметров. Согласно устоявшейся нотации, говорят, что слой свёртки использует фильтр W ×H, если каждый фильтр в этом слое имеет число — W ×H ×C, где, C размерность каналов в предыдущем слое.

Слой субдискретизации осуществляет уплотнение карт признаков предыдущего слоя и не изменяет количество карт. Каждая карта признаков слоя соединена с соответствующей картой признаков предыдущего слоя, каждый нейрон выполняет «сжатие» своего рецептивного поля посредством какой-либо функции.

Наиболее популярными видами этого слоя являются Max Pooling (из рецептивного слоя выбирается максимальное значение), Average Pooling (выбирается среднее значение) и L2 Pooling (выбирается норма L2) [Li et al., 2016]. С помощью слоя субдискретизации достигается устойчивость к небольшим сдвигам входного изображения, а также уменьшается размерность последующих слоёв [Goodfellow et al., 2016].

Полносвязный слой — обычный скрытый слой многослойного перцептрона, соединённый со всеми нейронами предыдущего слоя.

Таким образом, на вход свёрточной нейронной сети подаётся

изображения, а на выходе получается класс, к которому принадлежит изобрежание.

2.1 Densely Connected Convolutional Networks

Последние исследования показали, что конволюционные сети могут быть обучены более глубоко, более точно и более эффективно, если они содержат более короткие связи между слоями, близкими к входу, и слоями, близкими к выходу. В данной работе мы принимаем это наблюдение и представляем плотную конволюционную сеть (DenseNet), которая соединяет каждый слой с каждым другим слоем по принципу feed-forward. В то время как традиционные L-слойные конволюционные сети имеют L соединений - по одному между каждым слоем и его преемником - наша сеть имеет L(L + 1)/2 прямых соединений. Для каждого слоя в качестве входных данных используются отображения элементов всех предыдущих слоев, а их собственные отображения элементов будут использоваться в качестве входных данных для всех последующих слоев. Плотные сети имеют несколько неоспоримых преимуществ: они облегчают проблему исчезновения градиента, улучшают распространение признаков, способствуют повторному использованию признаков и значительно сокращают количество параметров. Мы оценили предложенную нами архитектуру на четырех высококонкурентных эталонных задачах

распознавания объектов (CIFAR-10, CIFAR-100, SVHN и ImageNet). denseNets достигает значительных улучшений в большинстве методов, при этом требуя меньше вычислений для достижения высокой производительности.

Основная проблема, рассматриваемая в статье, заключается в том, что при прохождении информации о входе или градиенте через множество слоев, она может исчезнуть и "смыться", когда достигнет конца (или начала) сети, т.е. проблема исчезновения градиента.

В статье предлагается архитектура, которая переводит это понимание в простую модель подключения: чтобы обеспечить максимальный поток информации между слоями в сети, мы подключаем все слои (с соответствующими размерами карты характеристик) непосредственно друг к другу. Для поддержания свойств feed-forward каждый слой получает дополнительные входные данные от всех предыдущих слоев и передает свою карту признаков всем последующим слоям. На рисунке 3 схематично показана эта схема.

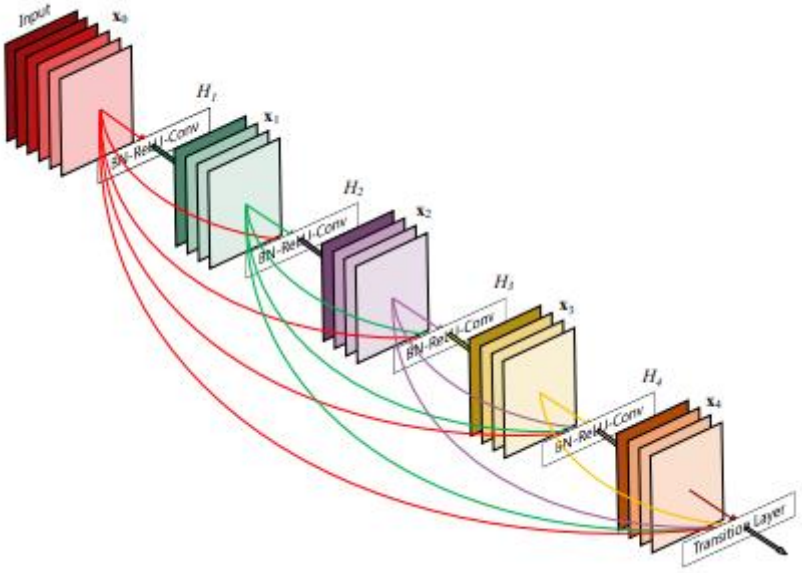

Рис 3 A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input.

DenseNets не нужно повторно изучать больше карт объектов и, следовательно, требуется меньше параметров, чем в традиционных сверточных сетях. Это подтверждается экспериментальными результатами на четырех эталонных наборах данных (CIFAR-10, CIFAR-100, SVHN и ImageNet), модели, используемые в этой статье, как правило, требуют меньше параметров и имеют сравнимую точность с существующими алгоритмами. Кроме того, модель значительно превосходит текущие самые современные результаты в большинстве тестовых задач.

Method	Depth	Params	C10	C10+	C100	C100+	SVHN
Network in Network [22]	-	-	10.41	8.81	35.68	-	2.35
All-CNN [32]	-	-	9.08	7.25	-	33.71	-
Deeply Supervised Net [20]	-	-	9.69	7.97	-	34.57	1.92
Highway Network [34]	-	-	-	7.72	-	32.39	-
FractalNet [17]	21	38.6M	10.18	5.22	35.34	23.30	2.01
with Dropout/Drop-path	21	38.6M	7.33	4.60	28.20	23.73	1.87
ResNet [11]	110	1.7M	-	6.61	-	-	-
ResNet (reported by [13])	110	1.7M	13.63	6.41	44.74	27.22	2.01
ResNet with Stochastic Depth [13]	110	1.7M	11.66	5.23	37.80	24.58	1.75
	1202	10.2M	-	4.91	-	-	-
Wide ResNet [42]	16	11.0M	-	4.81	-	22.07	-
	28	36.5M	-	4.17	-	20.50	-
with Dropout	16	2.7M	-	-	-	-	1.64
ResNet (pre-activation) [12]	164	1.7M	11.26*	5.46	35.58*	24.33	-
	1001	10.2M	10.56*	4.62	33.47*	22.71	-
DenseNet ($k = 12$)	40	1.0M	**7.00**	5.24	**27.55**	24.42	1.79
DenseNet ($k = 12$)	100	7.0M	**5.77**	**4.10**	**23.79**	**20.20**	1.67
DenseNet ($k = 24$)	100	27.2M	**5.83**	**3.74**	**23.42**	**19.25**	**1.59**
DenseNet-BC ($k = 12$)	100	0.8M	**5.92**	4.51	**24.15**	22.27	1.76
DenseNet-BC ($k = 24$)	250	15.3M	**5.19**	**3.62**	**19.64**	**17.60**	1.74
DenseNet-BC ($k = 40$)	190	25.6M	-	**3.46**	-	**17.18**	-

Рис 4 Частота ошибок в наборах данных CIFAR и SVHN

2.2 An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Хотя архитектура Transformer стала стандартом де-факто для задач обработки естественного языка, ее использование в компьютерном зрении все еще ограничено. В зрении внимание либо используется вместе со сверточной сетью, либо заменяет определенные компоненты сверточной сети, сохраняя при этом ее общую структуру. В статье показано, что такая зависимость от CNN является излишней и что чистые преобразователи, применяемые непосредственно к последовательностям блоков изображения, могут очень хорошо справляться с задачами классификации изображений. При предварительном обучении на больших объемах данных и передаче множества средних и малых эталонов распознавания изображений

(ImageNet, CIFAR-100, VTAB и т.д.), Vision Transformer (ViT) показывает отличные результаты по сравнению с современными сверточными сетями, требуя при этом меньше вычислительных ресурсов для обучения.

Статья делит изображение на блоки фиксированного размера и линейно встраивает каждый блок, Добавьте позиционные вложения и передайте полученную векторную последовательность стандартному преобразователю. Кодер. Для выполнения классификации в статье используются стандартные методы, добавляющие дополнительные обучаемые элементы. «Маркер классификации» для последовательности. Модель показана на рисунке 5.

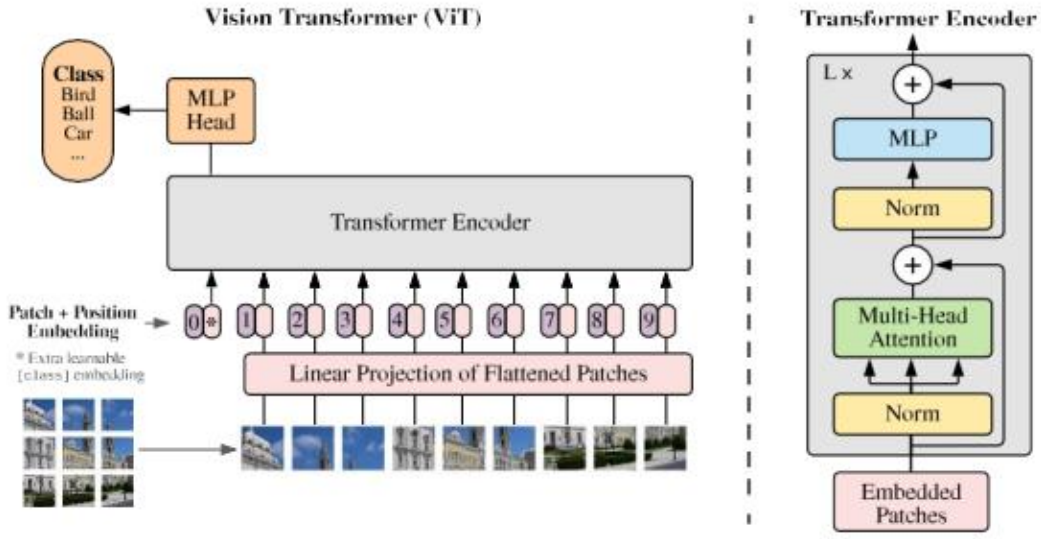

Рис 5 Обзор модели

Стандартный преобразователь получает на вход одномерную последовательность вложений токенов. Для обработки 2D-изображений мы преобразуем изображение $x \in R^{H \times W \times C}$ в плоскую 2D-

последовательность ES $x_P \in R^{N\times(P^2\cdot C)}$, где (H,W) — разрешение исходного изображения, а C — разрешение исходного изображения. количество каналов, (P, P) — разрешение каждого фрагмента изображения, а $N=HW/P^2$ — количество результирующих срезов, которое также служит эффективной длиной входной последовательности для преобразователя. Преобразователь использует постоянный размер скрытого вектора D во всех слоях, поэтому мы сглаживаем и сопоставляем патч с размером D с обучаемой линейной проекцией (уравнение 1). Мы называем результат этой проекции встраиванием больших двоичных объектов. Подобно токену Берта [Class], мы задаем обучаемое вложение для последовательности вложенных срезов ($Z^0_0=X_{Class}$), состояние которых на выходе преобразователя-кодировщика (Z0l) представлено как изображение y (уравнение 4). Как во время предварительной подготовки, так и во время тонкой настройки головка классификации присоединяется к Z^0_L, а головка классификации реализуется MLP с одним скрытым слоем во время предварительной подготовки и одним линейным слоем во время тонкой настройки. Добавьте позиционные вложения во вложения больших двоичных объектов, чтобы сохранить информацию о позициях. Мы используем стандартные обучаемые одномерные позиционные вложения, поскольку мы не наблюдаем значительного прироста производительности при использовании

более продвинутых двумерных позиционных вложений. Полученная последовательность векторов встраивания используется в качестве входных данных для кодировщика. Кодер-трансформер (Vaswani et al., 2017) состоит из чередующихся слоев многоголовочного самоконтроля (MSA) и блоков MLP (уравнения 2, 3). LayerNorm (LN) применяется перед каждым блоком, а остаточные соединения применяются после каждого блока (Wang et al., 2019; Baevski & Auli, 2019). MLP содержит два слоя с нелинейностью Gelu:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D},\ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \tag{4}$$

Рис 6 уравнения

В статье используется набор данных ILSVRC-2012 ImageNet, содержащий 1 тыс. классов и 1,3 млн изображений.

В документе оцениваются возможности обучения представлению ResNet, Vision Transformer (ViT) и Hybrid. Результаты показаны на рисунке 7.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Рис 7 Результат

Мы сообщаем среднее значение и стандартное отклонение точности,

усредненные по трем прогонам тонкой настройки. Зрение Модели трансформаторов, предварительно обученные на наборе данных JFT-300M, превосходят базовые модели на основе ResNet по всем параметрам. наборы данных, при этом для предварительной подготовки требуется значительно меньше вычислительных ресурсов. ViT предварительно обучен на меньший общедоступный набор данных ImageNet-21k также работает хорошо. *Слегка улучшенный результат 88,5% сообщается в Тувроне и др. (2020). По сравнению с современным уровнем техники объем вычислений, необходимых для предварительного обучения модели, по-прежнему значительно сокращается. Однако отметим, что на эффективность предварительной тренировки может влиять не только выбор архитектуры, но и другие параметры, такие как расписание обучения, оптимизатор, уменьшение веса и т. д. Наконец, общедоступный набор данных ImageNet-21k также хорошо работает на большинстве наборов данных, в то время как предварительное обучение требует меньше ресурсов: его можно обучить примерно за 30 дней с использованием стандартного облачного TPUv3 с 8 ядрами.

3. практический этап

Для статьи «Сверточные сети с высокой плотностью соединений» сопутствующий код показан на рисунке ниже.

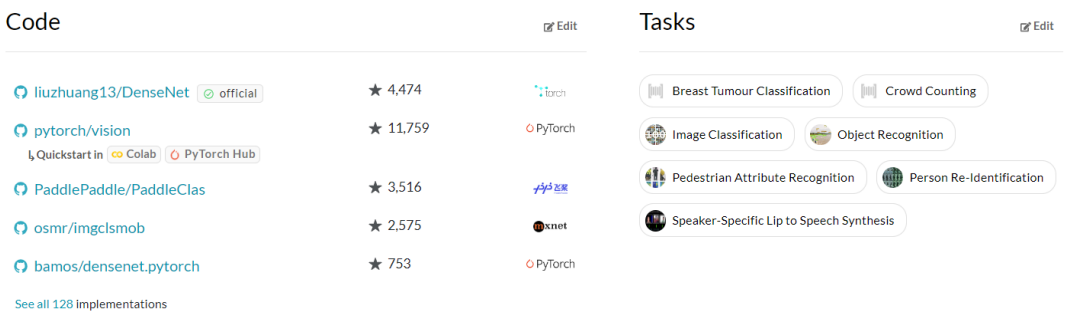

Рис 8 Код статьи

Мы переходим ко второму коду, поскольку он предоставляет ссылку на google colab.

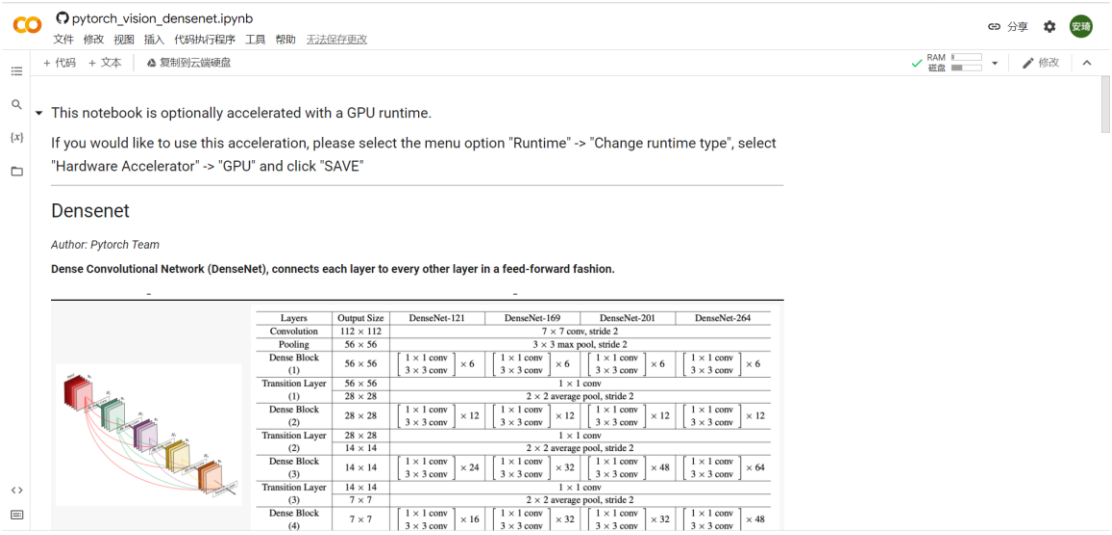

Рис 9 Интерфейс кода

Для статьи «Изображение стоит 16x16 слов: трансформеры для распознавания изображений в масштабе» сопутствующий код показан на рисунке ниже.

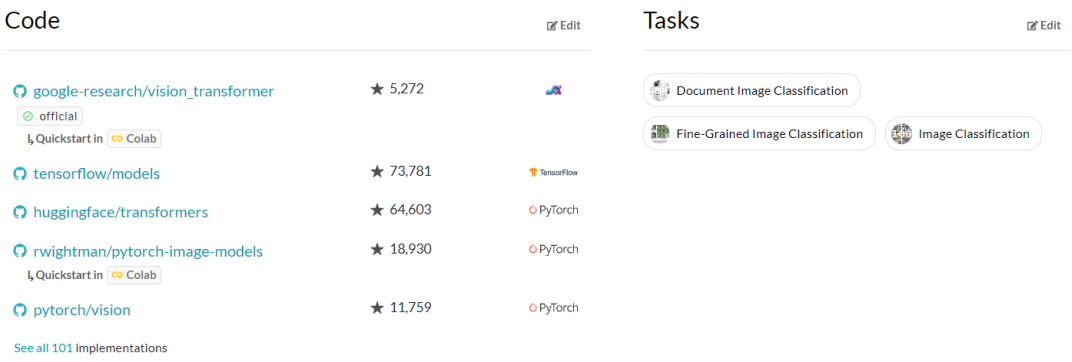

Рис 10 Код статьи

Переходим к коду.

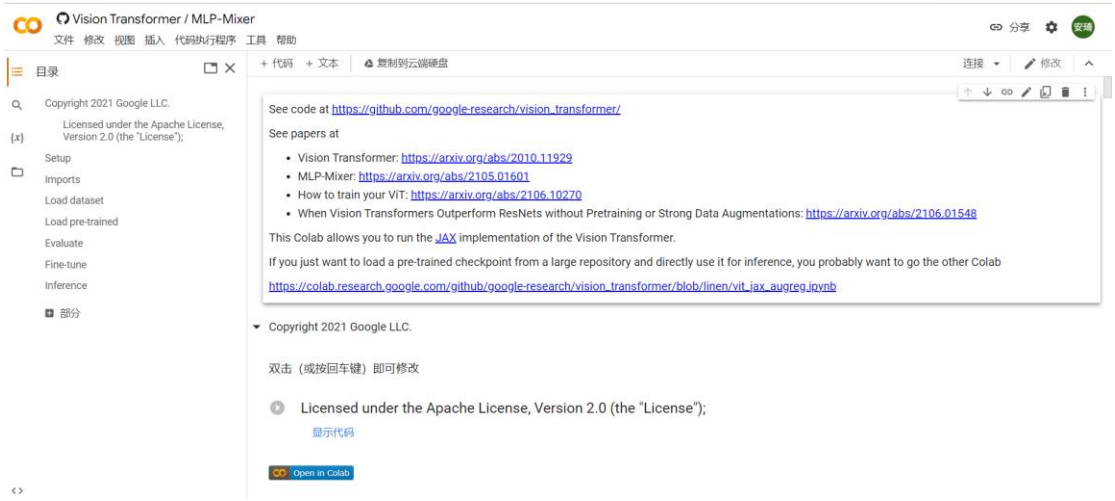

Рис 11 Интерфейс кода

Время выполнения и результаты были загружены на github.

4. Заключение

В течение семестра мы узнали кое-что о классификации изображений и некоторых простых методах повышения точности работы сверточных нейронных сетей, таких как улучшение изображения и регуляризация. Выполняя это задание, мы познакомились с новыми сетевыми архитектурами и завершили проверку результатов, используя уже написанный код. На практике, чтобы заставить код работать, пришлось многое подправить, и только от мелких исправлений у меня сильно болела голова. Было здорово иметь возможность самостоятельно разрабатывать архитектуру и писать код. Я буду продолжать упорно работать в ближайшие годы, чтобы стать успешным архитектором самостоятельно.

Список использованных источников

[1] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708). https://paperswithcode.com/paper/densely-connected-convolutional-networks

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. https://paperswithcode.com/paper/an-image-is-worth-16x16-words-transformers-1

[3] Сикорский, О. С. (2017). Обзор свёрточных нейронных сетей для задачи классификации изображений. *Новые информационные технологии в автоматизированных системах*, (20), 37-42.

[4] 【Технический обзор】Вы действительно понимаете классификацию изображений? https://zhuanlan.zhihu.com/p/47281243

[5] Машинное зрение (8) Классификация изображений https://zhuanlan.zhihu.com/p/157522722