**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 2](#_Toc189870353)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc189870354)

[1.1. Понятие свёрточных нейронных сетей 5](#_Toc189870355)

[1.2. Трансферное обучение 8](#_Toc189870356)

[1.2.1. Архитектура VGG 9](#_Toc189870357)

[1.2.2. Архитектура ResNet 10](#_Toc189870358)

[1.2.3 Архитектура MobileNet 11](#_Toc189870359)

[1.2.4. Архитектура EfficientNet 13](#_Toc189870360)

[СПИСОК ЛИТЕРАТУРЫ 16](#_Toc189870361)

ВВЕДЕНИЕ

Современная авиация играет ключевую роль в глобальной транспортной системе, обеспечивая быстрое перемещение людей и грузов на большие расстояния. С развитием технологий разнообразие воздушных судов непрерывно растёт: от пассажирских лайнеров и грузовых самолётов до специализированных моделей для военных и исследовательских задач. Точная идентификация типа воздушного судна по внешним признакам критически важна для решения таких задач, как управление воздушным движением, мониторинг безопасности, логистическое планирование и анализ спутниковых данных.

В последние годы значительно возрос интерес к использованию методов компьютерного зрения для автоматического анализа аэрокосмических изображений. Такие технологии применяются в гражданской и военной сферах, включая мониторинг аэропортов, контроль воздушного движения, автоматическое обнаружение объектов на спутниковых снимках и идентификацию моделей воздушных судов. Современные алгоритмы машинного обучения, такие как свёрточные нейронные сети, демонстрируют высокую эффективность [1] в задачах классификации изображений, но требуют дальнейшего совершенствования при применении к сложным сценам с высоким уровнем шума и разнообразием классов объектов.

Актуальность разработки надёжных методов машинного обучения для классификации воздушных судов обусловлена растущим объёмом спутниковых данных и необходимостью их оперативного анализа. Основная сложность заключается в вариативности ракурсов съёмки, влиянии погодных условий, наличии помех и высокой степени схожести между различными моделями воздушных судов. Совершенствование методов автоматической классификации позволит повысить точность идентификации и сократить нагрузку на специалистов, работающих с аэрокосмическими данными.

Целью данной работы является разработка системы классификации типов воздушных судов по спутниковым изображениям на основе свёрточной нейронной сети.

Для реализации поставленной цели были решены следующие задачи:

1. Определить наиболее подходящие модели свёрточных нейронных сетей для реализации трансферного обучения;
2. Определить метрики для объективного сравнения моделей;
3. Обучить модели на наборе данных спутниковых изображений воздушных судов;
4. Определить наиболее оптимальную модель для заданной цели;
5. Реализовать пользовательский интерфейс.

Объектом исследования является процесс автоматической классификации воздушных судов по изображениям; предметом исследования является применение свёрточных нейронных сетей для решения задачи классификации типов воздушных судов.

Методологической основой работы является использование методов глубокого обучения, в частности, свёрточных нейронных сетей и обучения с переносом для обработки и классификации изображений. В качестве инструментов использовались язык программирования Python, в частности библиотеки TensorFlow и Keras для реализации моделей машинного обучения, а также набор данных спутниковых снимков воздушных судов. Для разработки пользовательского интерфейса применялась библиотека PythonQt, которая позволила создать удобное и эффективное графическое представление результатов классификации.

Практическая значимость работы заключается в разработке системы, способной автоматически идентифицировать типы воздушных судов, что может быть применимо в задачах мониторинга, обороны и авиационной аналитики.

Структура работы включает введение, теоретический обзор современных методов классификации изображений с применением глубокого обучения, описание используемых моделей и методологии экспериментов, результаты тестирования разработанной системы, анализ их эффективности и заключение.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Понятие свёрточных нейронных сетей

Свёрточные нейронные сети (далее - СНС) являются классом алгоритмов машинного обучения, в основе которых лежит способность автоматически извлекать признаки из входных данных, что особенно эффективно при обработке изображений и видео. Это достигается благодаря применению свёрточных слоев, которые фильтруют информацию и сохраняют только наиболее значимые признаки [2].

Несмотря на непрерывную эволюцию СНС, набирающую все большие темпы за последнее десятилетие, основные части алгоритма остаются неизменными. Первая модель свёрточной нейронной сети – LeNet, разработанная в 1998 году Яном Лекуном и Леоном Ботту для задачи распознавания рукописных цифр и букв с помощью классификации [3], использует в своей основе те же (но модифицированные) компоненты, как и выпущенная в 2021 году модель СНС EfficientNetV2-L с более чем 120 миллионами параметров [4].

Архитектуру простейшей СНС можно разбить на 2 основные части, связанные друг с другом и идущие последовательно [3]:

1) Модуль извлечения признаков, состоящий из набора последовательных слоёв свёртки, слоёв активации и слоёв подвыборки.

2) Обучаемый классификационный модуль, представляющий из себя полносвязную нейронную сеть.

Свёрточные слои — основной компонент СНС, обрабатывающий входные данные с помощью фильтров (ядер свёртки). Свёртка позволяет сети выявлять локальные признаки, такие как края, текстуры и другие элементы изображения. Ядра свертки - матрицы весов небольшого размера, как правило, 3x3. Фильтр перемещается по входному признаку, например, градиенту яркости пикселей изображения, и вычисляет свёртку, то есть скалярное произведение значений входных пикселей и весов ядра. В результате получается карта признаков, содержащая информацию о выявленных структурах. Во время обучения нейронной сети градиентный спуск изменяет веса фильтров, чтобы сеть лучше выделяла значимые признаки. На рисунке 1 представлено визуальное представление работы ядра свертки [2].

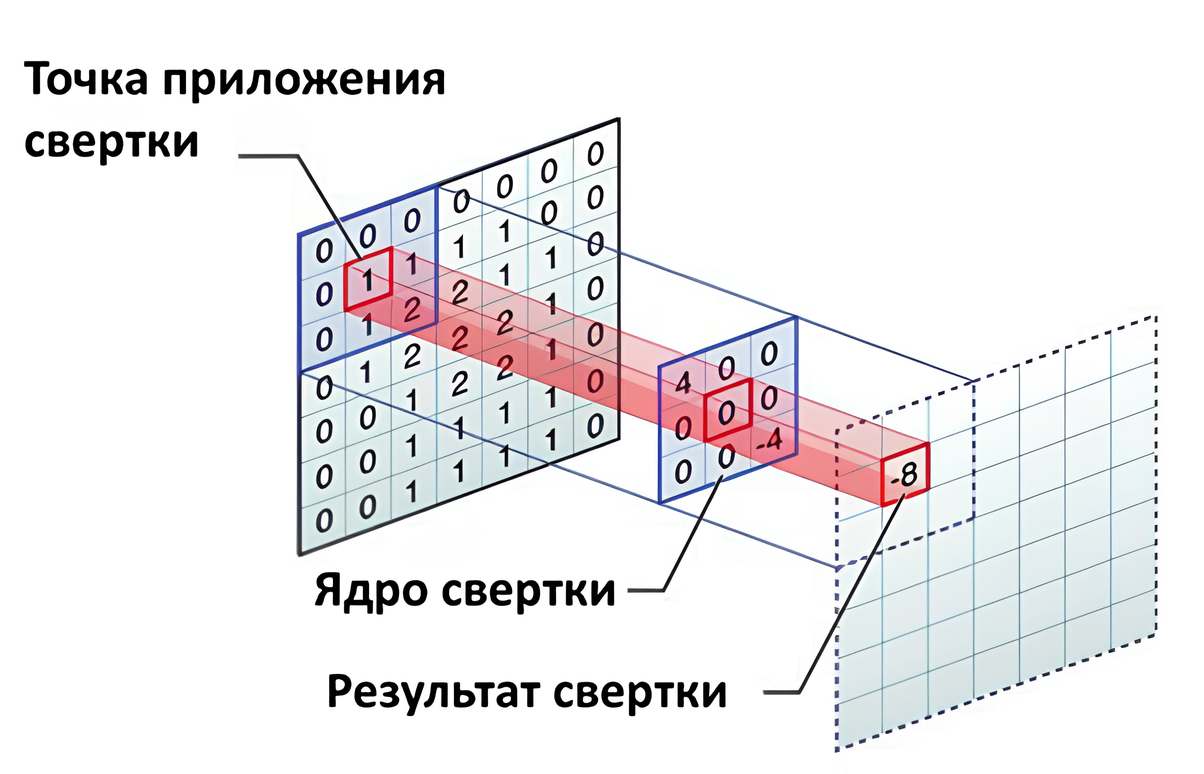


Рисунок 1 – Визуальное представление работы ядра свертки

После применения сверточных слоёв обычно используется слой активации для введения нелинейности в модель. Это позволяет сети моделировать сложные зависимости между признаками и улучшает её способность к обобщению. Одной из наиболее распространённых функций активации является ReLU (Rectified Linear Unit), которая заменяет все отрицательные значения на ноль, сохраняя положительные значения без изменений. Другие популярные функции активации включают сигмоидную функцию и гиперболический тангенс (tanh), каждая из которых имеет свои особенности и области применения [5].

Слои подвыборки, в свою очередь, позволяют снизить пространственное разрешение данных, уменьшая размерность и вычислительные затраты, а также способствуют инвариантности к небольшим изменениям в изображении. Существуют две разновидности слоев подвыборки: максимальная подвыборка (max pooling) и средняя подвыборка (average pooling), где используется максимальное и среднее значение признаков соответственно [6]. На рисунке 2 представлено визуальное представление работы слоев максимальной и средней подвыборки.

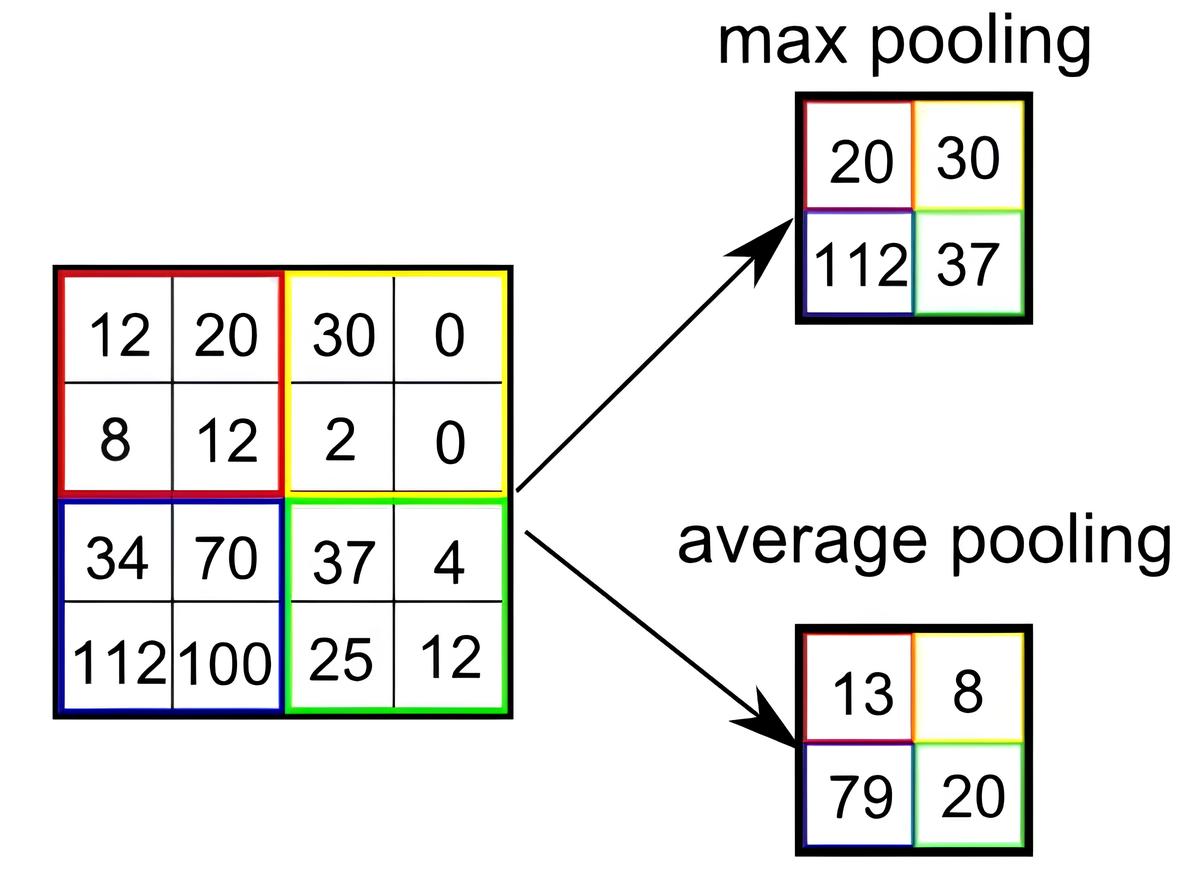


Рисунок 2 – Визуальное представление работы слоев максимального и среднего объединения

После этапов свёртки и объединения полносвязный слой выполняет интеграцию этих признаков для принятия окончательного решения о принадлежности объекта к определённому классу. Полносвязный слой преобразует входной вектор выделенных признаков в выходной вектор вероятностей принадлежности классифицируемого объекта к определённому классу. В задачах классификации на последнем полносвязном слое часто используется функция активации softmax, которая нормализует выходные значения в диапазон от 0 до 1, представляя собой вероятности.

На рисунке 3 представлена стандартная архитектура простейшей СНС, где «скрытые слои» - модуль выделения признаков, а «классификация» - полносвязная нейронная сеть – классификатор.

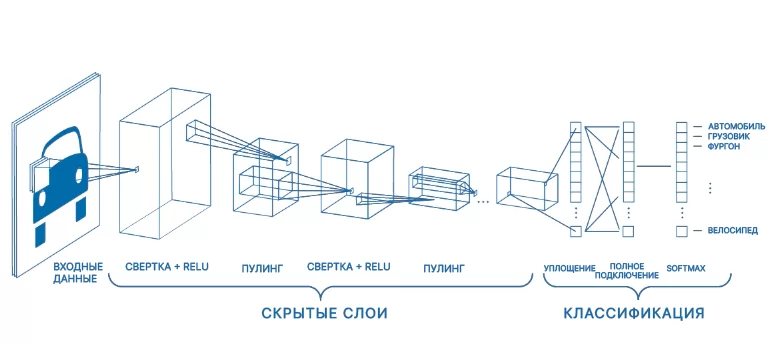


Рисунок 3 – Стандартная архитектура простейшей СНС

1.2. Трансферное обучение

Современные СНС обладают высокой эффективностью в задачах классификации изображений, однако их обучение «с нуля», а именно обучение как модуля извлечения признаков, так и классификационного модуля, требует значительных вычислительных ресурсов и больших объемов данных. Например, обучение модели AlexNet, которая произвела революцию в области машинного обучения в 2012 году, потребовало использования нескольких графических процессоров и заняло значительное время [7].

Для преодоления этих ограничений применяется метод трансферного обучения. Этот подход позволяет использовать веса, полученные моделью при решении одной задачи, для ускорения и улучшения обучения в другой, схожей задаче. В частности, модели, обученные на обширном наборе данных ImageNet Large Scale Visual Recognition Challenge (далее - ILSVRC), содержащем 1,461,406 размеченных изображений [8], широко используются в качестве основы для решения специализированных задач.

В рамках данной работы используется метод трансферного обучения, при котором веса предобученной модели, обученной на наборе данных ILSVRC, «замораживаются». Это означает, что значения весов в слоях модели-основы остаются неизменными, и не подлежат обновлению в процессе обучения на новых данных. В результате обучение происходит лишь на добавленной части модели, которая отвечает исключительно за классификацию, используя признаки, извлечённые предобученной моделью.

В следующих разделах рассматриваются архитектуры предобученных моделей, к которым была применена методика трансферного обучения для последующего выбора наиболее оптимальной модели для решения задачи классификации воздушных средств на спутниковых изображениях.

1.2.1. Архитектура VGG

Архитектура VGG (Visual Geometry Group) была разработана А. Зиссерманом и К. Симоняном в 2014 году. Данная архитектура отличается простой и последовательной архитектурой, которая использует несколько блоков свёрток с маленькими ядрами (3×3), за которыми следуют слои подвыборки. Разработчики представили две версии моделей на основе этой архитектуры, отличающихся глубиной сети: VGG-16 и VGG-19 (16 и 19 слоев соответственно). Увеличение числа слоёв относительно менее глубоких архитектур позволило модели, использующей данную архитектуру, захватывать более сложные и абстрактные признаки из изображений, что способствовало повышению точности классификации [9]. На рисунке 4 изображено визуальное представление модели VGG-16.

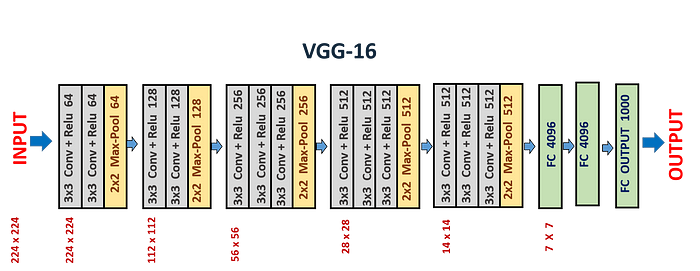


Рисунок 4 - Визуальное представление модели VGG-16

1.2.2. Архитектура ResNet

Архитектура ResNet (Residual Network) была разработана в 2015 году группой исследователей под руководством Кайминга Хэ. Данная архитектура была разработана для решения проблемы затухания градиентов, которая возникает при обучении глубоких нейронных сетей. Затухание градиентов приводит к тому, что градиенты ошибки становятся чрезвычайно малыми при обратном распространении, что затрудняет обновление весов в начальных слоях сети и замедляет процесс обучения [10].

В архитектуре ResNet используется метод, называемый "пропускные соединения", которые позволяют входным данным слоя напрямую передаваться в следующие слои, минуя один или несколько промежуточных слоёв. Преимущество использования пропускных соединений в том, что они помогают пропускать слои, которые ухудшают производительность сети. Это позволяет эффективно обучать очень глубокие нейронные сети, избегая проблем с затуханием градиентов. Данная техника позволила авторам ResNet реализовать СНС глубиной в 1202 слоя [11].

В оригинальной статье также представлены несколько версий моделей на основе ResNet, различающихся по глубине сети: ResNet-18, ResNet-34, ResNet-50, ResNet-101 и ResNet-152. Число после "ResNet-" указывает на количество слоёв в модели [11].

На рисунке 5 изображено визуальное представление архитектуры модели ResNet-34.

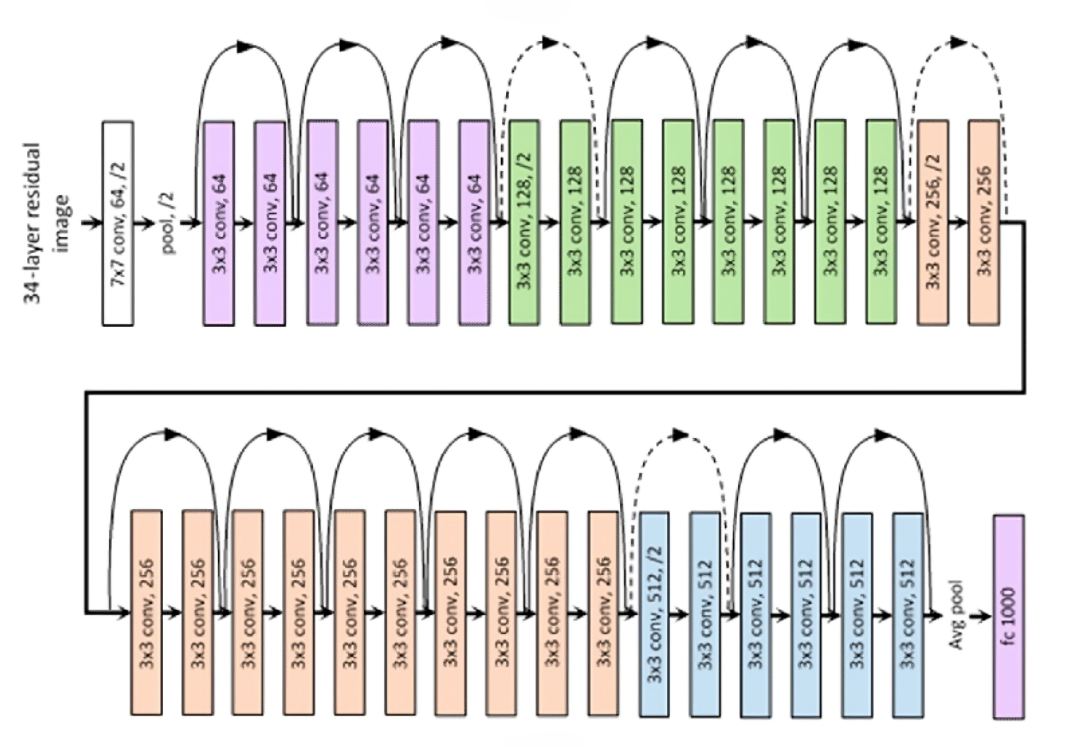


Рисунок 5 – Визуальное отображение архитектуры модели ResNet-34

1.2.3 Архитектура MobileNet

Архитектура MobileNet была разработана в 2017 году группой исследователей из компании Google включая Эндрю Г. Ховарда, Менглонга Чжу, Бо Чена и других. MobileNet проектировалась для работы в мобильных устройствах и встроенных системах и, как следствие, большой акцент при разработке уделялся производительности и эффективности. Ключевой особенностью MobileNet является использование глубинных раздельных сверток (depth-wise separable convolutions), которые позволяют значительно уменьшить количество параметров и вычислительные затраты без существенной потери точности [12].

В стандартной свёртке каждый фильтр применяется ко всем каналам входного изображения, что приводит к значительным вычислительным затратам. Глубинная раздельная свёртка разделяет этот процесс на два этапа [13]:

1. Глубинная свёртка (depth-wise convolution): Каждый фильтр применяется отдельно к каждому каналу входного изображения.

2. Точечная свёртка (point-wise convolution): 1×1 свёртка, объединяющая результаты предыдущего шага и смешивающая информацию между каналами.

На рисунке 6 представлено визуальное представление архитектуры модели на основе MobileNet.

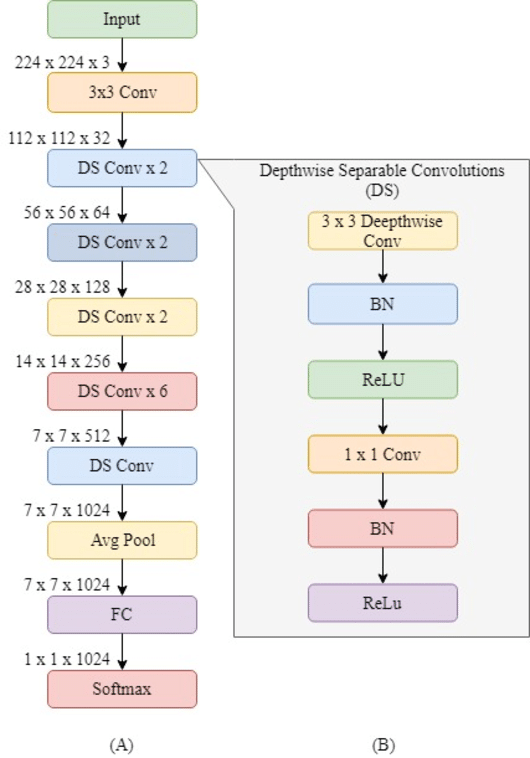


Рисунок 6 - Визуальное представление модели на основе архитектуры MobileNet

В дальнейшем были разработаны вторая (MobileNetV2) и третья (MobileNetV3) версии архитектуры. В MobileNetV2 использовались инвертированные остаточные блоки и линейные узкие места, которые позволили улучшить передачу информации и эффективность модели. MobileNetV3 была оптимизирована для мобильных устройств с помощью аппаратно-ориентированного поиска архитектур и включала такие нововведения, как функции активации h-swish и модули "squeeze-and-excitation", что позволило достичь лучшего баланса между точностью и производительностью [13].

1.2.4. Архитектура EfficientNet

Архитектура EfficientNet была разработана в 2019 году исследователями Google AI, включая Минсина Тана и Куока В. Ле. Ключевой особенностью EfficientNet является метод комплексного масштабирования, с помощью которого возможно одновременно увеличивать глубину (количество слоев), ширину (количество нейронов или фильтров в каждом слое) и разрешение (размер входных данных) сети с помощью единого коэффициента. Данный метод позволяет достичь оптимального баланса между точностью и вычислительными ресурсами, которые требуются для работы модели [14].

Помимо использования метода комплексного масштабирования, данную архитектуру выделяют особенные блоки свертки – MBConv (Mobile Inverted Bottleneck Convolution - обратная свёртка с инвертированным узким горлышком). Они представляют собой модификацию глубинных раздельных сверток (используемых, в архитектуре MobileNet) с использованием механизма «squeeze-and-excitation», который применяется после свёрточных операций. Данный механизм адаптивно взвешивает важность различных каналов и позволяет модели фокусироваться на наиболее важных признаках, подавляя менее важные [15].

Совместное использование комплексного масштабирования и блоков свёртки MBConv позволила моделям на основе EfficientNet достичь наилучших результатов в задаче классификации изображений на основе популярных датасетов, таких как ImageNet и CIFAR-100, при этом существенно снижая требования к памяти и скорости обработки по сравнению с предыдущими архитектурами CNN [16].

На рисунке 7 представлено визуальное представление модели на основе архитектуры EfficientNet.

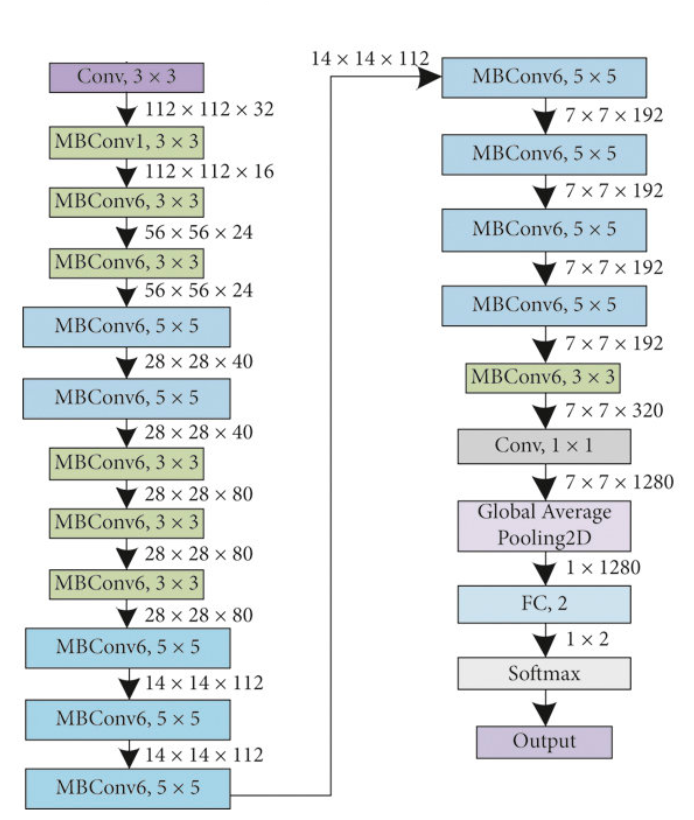
**

Рисунок 7 – Визуальное представление модели на основе архитектуры EfficientNet

В 2021 году была разработана усовершенствованная версия данной архитектуры - EfficientNetV2, особенностью которой было использование нового поколения свёрточных блоков - Fused-MBConv, а также автоматической генерации архитектуры с помощью алгоритма поиска нейронной архитектуры (Neural Architecture Search) [17].

СПИСОК ЛИТЕРАТУРЫ

1. Смирнов А. В., Иванов Е. С., Использование механизма сверточных нейронных сетей для поиска объектов на аэрофотоснимках // Программные системы: теория и приложения.  2017. Том 4. №35. C. 85–99.
2. Самое главное о нейронных сетях. Лекция в Яндексе // Хабр URL: https://habr.com/ru/companies/yandex/articles/307260 (дата обращения: 07.02.2025).
3. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-Based Learning Applied to Document Recognition // Proceedings of the IEEE. 1998. Том 86. №11. С. 2278–2324.
4. Tan M., Le Q.V. EfficientNetV2: Smaller Models and Faster Training // Proceedings of the 38th International Conference on Machine Learning. 2021. Том 139. С. 10096–10106.
5. Основные функции активации в нейронных сетях // YourToDo URL: https://yourtodo.life/ru/posts/osnovnyie-funktsii-aktivatsii-v-nejronnyih-setyah (дата обращения: 07.02.2025).
6. Понимание сверточных нейронных сетей через визуализации в PyTorch // Хабр URL: https://habr.com/ru/articles/436838/ (дата обращения: 07.02.2025).
7. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks // Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12). 2012. С. 1097–1105.
8. Russakovsky O., Deng J., Su H., Krause J., Satheesh S., ImageNet Large Scale Visual Recognition Challenge // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015. C. 3481–3489.
9. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // Proceedings of the International Conference on Learning Representations (ICLR 2015). 2015.
10. Исчезающий градиент // Ultralytics URL: https://www.ultralytics.com/ru/glossary/vanishing-gradient (дата обращения: 07.02.2025).
11. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
12. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // arXiv preprint arXiv:1704.04861. 2017. URL: <https://arxiv.org/abs/1704.04861>.
13. Understanding and Implementing MobileNetV3 // Medium URL: https://medium.com/@RobuRishabh/understanding-and-implementing-mobilenetv3-422bd0bdfb5a (дата обращения: 07.02.2025).
14. Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // Proceedings of the 36th International Conference on Machine Learning (ICML 2019). Том 97. С. 6105–6114.
15. What is EfficientNet? The Ultimate Guide. // Roboflow URL: https://blog.roboflow.com/what-is-efficientnet/?utm\_source (дата обращения: 08.02.2025).
16. Efficientnet Architecture // GeeksGorGeeks URL: https://www.geeksforgeeks.org/efficientnet-architecture/ (дата обращения: 08.02.2025).
17. Tan M., Le Q. EfficientNetV2: Smaller Models and Faster Training. Proceedings of the 38th International Conference on Machine Learning. 2021. Том 139. С. 10096–10106.