Исследование возможности классификации человеческих действий

И. Б. Алексеев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2311242@edu.misis.ru

П. Е. Злакоманов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2301834@edu.misis.ru

***Аннотация*— исследование направлено на понимание нейронными сетями человеческого поведения и присвоение класса каждому действию. Распознавание человеческих действий имеет широкий спектр применения и поэтому привлекает все большее внимание в области компьютерного зрения. Действия человека могут быть представлены с использованием различных модальностей данных, таких как RGB, скелет, глубина, инфракрасный порт, облако точек, поток событий, звук, ускорение, радар и сигнал Wi-Fi, которые кодируют различные источники полезной, но отдельной информации и имеют различные преимущества в зависимости от сценария и применения. В работе строятся модели классификации изображений с использованием CNN, которые классифицируют класс деятельности, выполняемый человеком на датасете с Kaggle и проверяется работа на реальных данных.**

Ключевые слова — Компьютерное зрение, Детекция человеческих действий, Распознавание человеческих действий, CNN, ResNet50, Xception, DenseNet169.

I. Введение

Нейронные сети нашли свое применение в разных отраслях. Например, в решениях проблем прогнозирования поведения транспортных средств на месте дорожного движения на основе анализа изображений. Для обнаружения объектов и оценки их местоположения используется 3D-детектор глубокой нейронной сети (DNN), где кинематическая модель велосипеда рассматривается как модель поведения, при которой каждое транспортное средство обрабатывается отдельно, без учета влияния других участников дорожного движения [1]. Также, в статье «Оценка точности трамвайной системы позиционирования в условиях высотного строительства с использованием данных визуальных геоинформационных систем» описан подход к оценке точности локализации трамвая, движущегося в городской среде, где трамвай должен быть локализован с точностью подметки. Поскольку информация GPS в городской среде не обеспечивает такого уровня точности, предлагается решение, основанное на использовании информации о системе зрения. Используя ключевые точки, можно оценить движение объекта между изображениями, полученными в разных проходах, в то время как это движение также можно рассчитать с помощью данных бортовой навигационной системы. Сопоставление этих перемещений позволяет оценить точность навигационной системы на борту [2]. В алгоритме автоматической посадки БПЛА с использованием компьютерного зрения рассматриваются алгоритмы систем зрения для автоматической посадки беспилотных летательных аппаратов (БПЛА). Представлены основные алгоритмы поиска вертолетной площадки и адаптации системы управления БПЛА к автономной посадке, рассмотрены алгоритмы решения навигационных задач, построения блоковых диаграмм автоматического управления. Разработанные алгоритмы позволяют реализовать автоматическую систему посадки для БПЛА с использованием систем технического зрения, с использованием различных параметров камеры, алгоритмов, позволяющих проводить исследования, в рамках автономной навигации беспилотных летательных аппаратов. В статье рассматривается разработанный алгоритм, позволяющий выделить определенные характерные точки для визуальной навигации. Кроме того, разработана система поиска характерных точек, позволяющая осуществлять автоматическую посадку БЛА. Эти алгоритмы могут быть полезны для автономных систем посадочных БЛА и для отслеживания траектории системы [3]. Аналогичная статья «Недорогая навигационная система для БПЛА» рассматривает малогабаритную навигационную систему NV-micro компании Integral Ltd, где представлены конструкция навигационной системы, алгоритм и результаты тестирования навигационной системы на световоммоторный самолет, демонстрирующий точность предлагаемой навигационной системы [4]. Виртуальное разворачивание или разворачивание, цифровое разворачивание, выравнивание или разворачивание - все эти термины используются для описания процесса выпрямления поверхности томографически реконструированного цифрового объекта. Цифровое выравнивание применяется в оптическом распознавании текста. В статье "От томографической реконструкции до автоматического распознавания текста: следующая задача для искусственного интеллекта" представлен открытый и кумулятивный набор данных CT-OPC-2022, который служит эталоном для реконструированных систем цифрового сплющивания и распознавания объектов [5].

Распознавание активности человека (HAR - Human Activity Recognition) - это известная исследовательская тема, которые используют камеры и микрофоны для записи движений тела: они не вмешиваются в личную жизнь пользователей, поскольку не включают видеозаписи в частных и домашних контекстах, менее чувствительны к окружающему шуму, дешевы и эффективны с точки зрения энергопотребления. Более того, широкое распространение встроенных сенсоров в смартфонах делает эти устройства повсеместными.

Одной из основных проблем в сенсорных методах HAR является представление информации. Традиционные методы классификации основаны на

признаках, которые созданы и извлечены из кинетических сигналов. Однако эти признаки выбираются в основном на эвристической основе, в соответствии с поставленной задачей. Часто процесс извлечения признаков требует глубоких знаний в области применения или человеческого опыта и все же это приводит только к поверхностным признакам. Типичные методы HAR плохо масштабируются для сложных паттернов движения и в большинстве случаев не показывают хороших результатов на динамических данных, то есть данных, полученных из непрерывных потоков.

В этой работе мы применяем сверточные нейронные сети ResNet50, Xception, DenseNet169, которые имеют разные архитектуры, для исследования возможностей классификации HAR и сравниваем результаты.

Мы классифицируем действия человека, используя различные базовые модели с трансферным обучением, что требует средних вычислительных мощностей. В настоящее время в свободном доступе есть много данных для обучения.

II. Наборы данных

Для обучения и тестирования рассматриваемых в данной работе нейросетей использовались открытый набор данных, и собранные авторами. Рассмотрим используемый открытый набор.

A. *Kaggle HAR*

Мы взяли набор данных с Kaggle, который содержит:

- 15 различных классов человеческих действий.
- Около 12 тысяч помеченных изображений, включая изображения для валидации.

Каждое изображение относится только к одной категории человеческой активности и сохранено в отдельных папках для каждого помеченного класса.

На рисунке 1 представлены распределение классов активности:

Рисунок 1. Распределение классов активности

Рисунок 2. Примеры изображений

III. Нейросетевые архитектуры

A. *DenseNet169*

DenseNet представляет собой инновационную нейронную сеть, основанную на концепции skip connection. Структура DenseNet начинается с входного сверточного слоя, за которым следует блок DenseBlock. После этого принцип повторяется. Входные карты активации передаются каждому слою в блоке, обеспечивая плотное соединение информации.

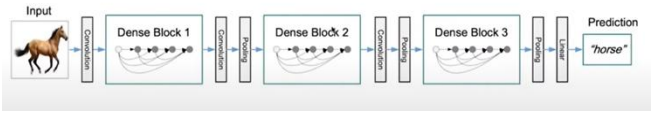

Рисунок 3. Структура DenseNet

Каждый блок DenseNet функционирует следующим образом: первый сверточный слой выдает карты активации, которые передаются следующему слою. Затем второй слой получает карты активации от обоих предыдущих слоев и выдает увеличенное количество карт. Процесс повторяется, увеличивая количество передаваемых карт активации с каждым последующим слоем.

Преимуществом DenseNet является высокий градиентный поток (strong gradient flow), что содействует борьбе с затуханием градиентов. Это позволяет создавать глубокие сети, например, DenseNet-264.

Кроме того, благодаря особенностям передачи информации между слоями, DenseNet эффективна в обучении, даже на небольших наборах данных.

Так как каждый сверточный слой внутри блока учитывает информацию из всех предыдущих слоев, сеть способна выделять разнообразные фичи. Нижние слои принимают во внимание более простые паттерны из верхних слоев, что может быть полезно для детекции низкоуровневых паттернов. Это делает DenseNet более эффективной на малых наборах данных.

B. ResNet

ResNet является сверточной нейронной сетью, используем ту, что содержит в себе 50 слоёв. На рисунке 4 представлена общая архитектура нейронной сети ResNet50.

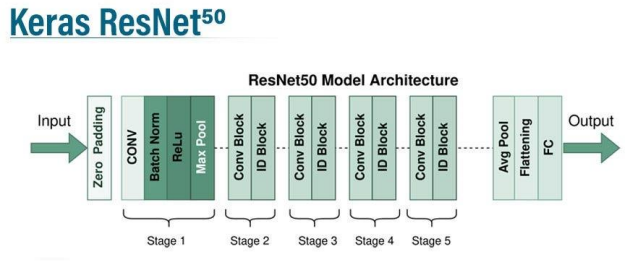

Рисунок 4. Архитектура ResNet50

После первого слоя и пулинга начинаются ResNet блоки, представленные на Рисунке 4. ResNet Block — это блок внутри Skip Connection, состоящий из двух слоев сети. На Рисунке 5 представлен пример ResNet Block.

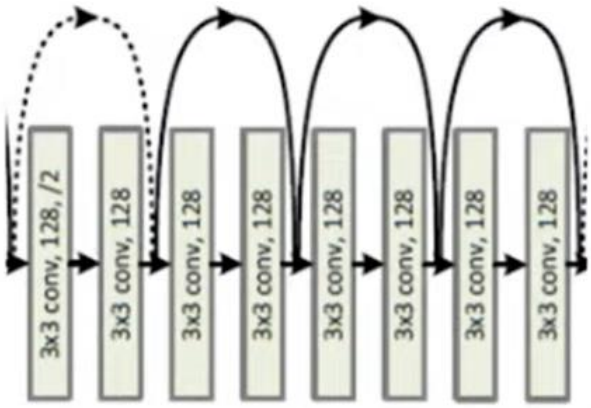

Рисунок 5. Пример ResNet Block

Пример ResNet Block включает 6 слоев, выдающих по 64 активации, затем 8 слоев с 128 активациями, и так далее.

Residual Blocks представляют ключевой элемент архитектуры ResNet, играя важную роль в обеспечении эффективности и глубины нейронной сети.

В каждом Residual Block сети ResNet присутствуют ровно два весовых слоя. Однако различия заключаются в том, как эти веса добавляются и взаимодействуют с Batch Normalization и ReLU.

ResNet одна из наиболее успешных архитектур в решении задач классификации изображений, поэтому решили использовать её тоже.

C. Xception

Inception схожа с концепцией ResNet. Вместо последовательных сверточных слоев в Inception используются несколько параллельных путей, основанных на сверточных слоях. Xception, представленная в 2017 году Франсуа Шолле и его командой, представляет собой эволюцию InceptionV3. Основная идея Xception заключается в полном пересмотре архитектуры Inception, заменив сверточные слои на глубокие разветвленные блоки глубокого разложения. Этот подход позволил модели эффективнее использовать параметры и повысить ее обобщающую способность. Содержит в себе 71 слой.

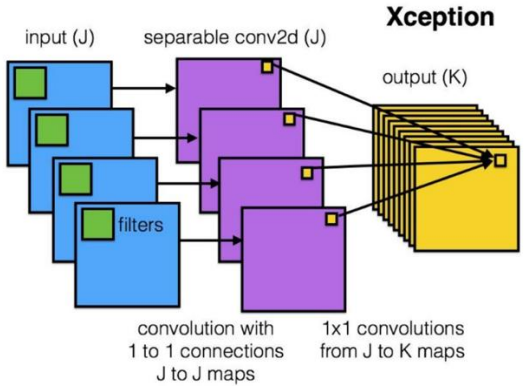

Рисунок 6. Пример Xception-модуль

На рисунке 6 и 7 представлен Xception-модуль и архитектура соответственно.

Данные сначала проходят через входной поток, затем через средний поток, который повторяется восемь раз, и, наконец, через выходной поток. Все слои Convolution и SeparableConvolution сопровождаются пакетной нормализацией.

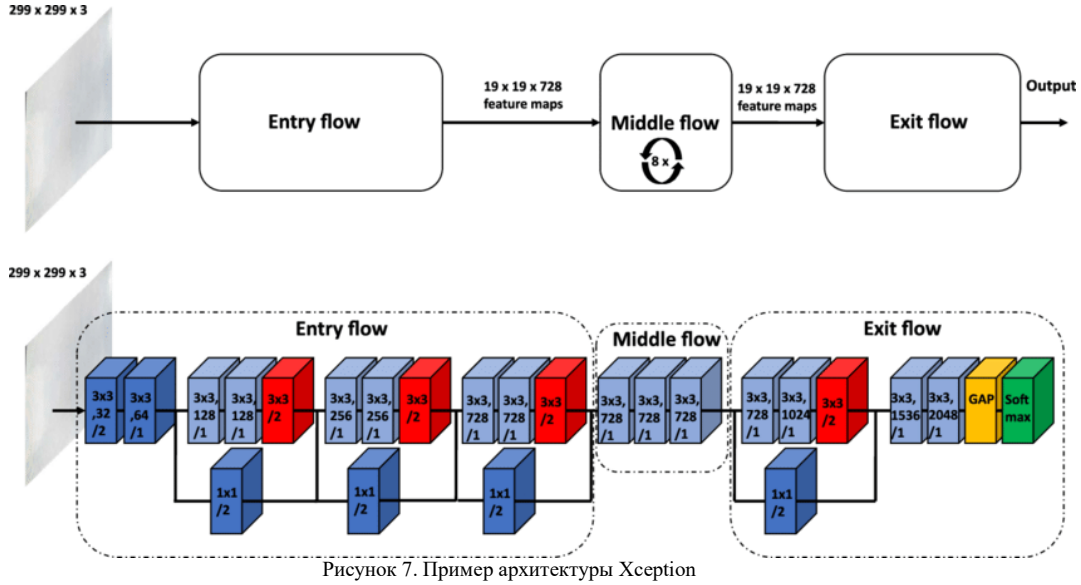

Рисунок 7. Пример архитектуры Xception

IV. Сравнение

Мы использовали такой подход, который включает в себя обучение 3 нейронных сетей с разными архитектурами и количеством слоёв на данных с Kaggle, а потом тестируем обученную модель на своём маленьком датасете, в котором преимущественно реальные данные.

Стоит отметить, что при обучении мы используем оптимизирующую функцию Adam. Основной идеей Adam является комбинация методов Momentum и RMSprop. Этот оптимизатор подстраивается под различные требования обучения, обеспечивая баланс между скоростью сходимости и адаптивностью к разным параметрам.

Adam поддерживает две переменные момента: первый момент (по аналогии с Momentum) и второй момент (по аналогии с RMSprop). Эти переменные вычисляются для каждого параметра модели.

Обновление первого момента (m): Отражает скорость изменения параметра

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times \nabla J_t$$

Обновление второго момента (v): Хранит информацию о квадрате градиента.

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times (\nabla J_t)^2$$

Коррекция смещения (bias correction): Учитывает начальные шаги оптимизации.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Обновление параметра (θ): Применяется для обновления весов модели.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \times \hat{m}_t$$

Где:

- ∇Jt - градиент функции потерь по параметру на шаге t
- β1 и β2 - коэффициенты затухания моментов
- η - шаг обучения
- ϵ - маленькое число для численной стабильности

Начинали мы с ResNet, т.к. она имеет наименьшее количество слоёв. Использовали тренированные веса Imagenet для каждой нейронной сети, каждая обучалась в 20 эпох.

ТАБЛИЦА I. Оценка точности и потерь на данных Kaggle после обучения

	ResNet50	Xception	DenseNet169
ValAccuracy	68%	74%	77%
TrainAccuracy	71%	77%	83%
TrainLoss	1.101	0.76	0.6
ValLoss	1.498	0.865	0.843
Recall	74%	74%	86%
Precision	87%	90%	88%

DenseNet169 демонстрирует лучшую производительность с точки зрения точности и потерь на валидационных данных, у неё самая высокая доля правильно классифицированных положительных случаев от общего числа предсказанных положительных случаев, т.к. у неё больше слоёв и она может увидеть больше признаков, в то время как ResNet50 проявляет наименьшую производительность среди рассматриваемых моделей, так же у неё наибольшие потери, что может указывать на то, что данные для неё слишком сложные. Xception находится между ними по производительности, обеспечивая умеренные результаты как по точности, так и по потерям, что является хорошим результатом, учитывая, что у неё 71 слой.

Чтобы проверить работу обученных моделей на реальных данных, мы собрали свой датасет, который содержит реальные изображения, сделали его разметку и взяли немного изображений с датасета Kaggle, примеры изображений на 8 рисунке.

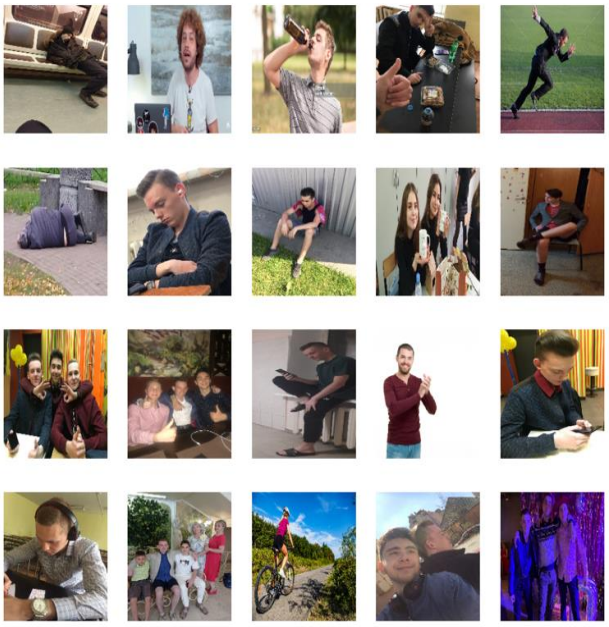

Рисунок 8. Примеры изображений

ТАБЛИЦА II. Результаты на реальных данных

	ResNet50	**Xception**	**DenseNet169**
Accuracy	13%	27%	27%

По результатам видно, что модели справились очень плохо, причём если изучить какие метки поставила сеть изображению, то видно, что реальные изображения он классифицирует, по большей части, ошибочно, а изображения с датасета Kaggle по большей части, правильно.

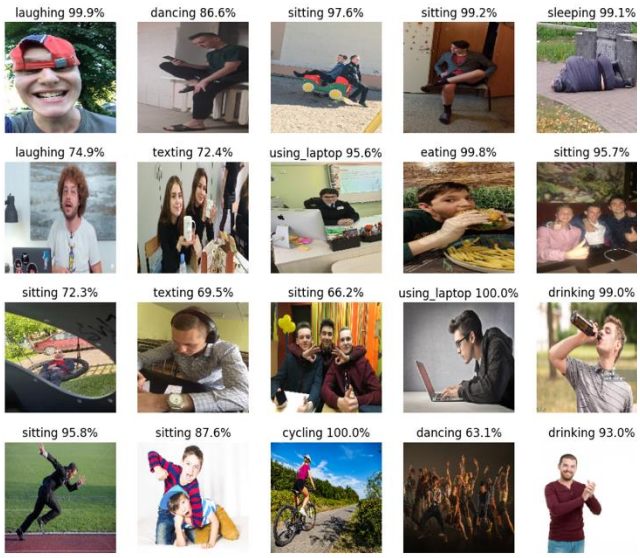

Рисунок 9. Примеры изображений с классификацией

В рисунке 9 представлены результаты классификации нейронной сети DenseNet169, у нашего датасета есть проблема с разметкой, потому что действие на одном изображении можно классифицировать по-разному, например, на одном изображении можно увидеть, что люди и сидят, и смеются, и обнимаются, если брать во внимание этот факт, то точность получается в районе 70%, касается DenseNet и Xception, модели более уверенно справляются со своими данными, потому что они более однозначные, т.е. простые. В итоге, можно сказать, что результат не такой уж и плохой, как кажется, но датасет с Kaggle слишком простой, чтобы использовать тренированные на нём модели где-то ещё.

V. Заключение

В заключение, модели были обучены на разнообразных данных, включая открытый набор Kaggle HAR, с последующим сравнением их результатов.

Результаты обучения на данных Kaggle показали, что DenseNet169 проявила лучшую производительность по сравнению с ResNet50 и Xception. Она продемонстрировала высокую точность классификации и более низкие потери как на обучающем, так и на валидационном наборах данных. С другой стороны, ResNet50 показала наименьшие результаты, что может быть связано с сложностью предоставленных данных.

Однако, при тестировании обученных моделей на собранном датасете реальных изображений, столкнулись с низкими результатами всех моделей. Вероятная причина заключается в неоднозначной разметке и сложности классификации действий на реальных изображениях.

Важно отметить, что полученные результаты подчеркивают важность корректной разметки данных и подготовки реальных датасетов для тестирования моделей. Несмотря на относительно высокую производительность на данных Kaggle, модели требуют доработки и настройки для более точного распознавания человеческих действий в реальных условиях.

Таким образом, дальнейшие исследования должны уделить внимание улучшению разметки данных, а также оптимизации параметров моделей для повышения их обобщающей способности на реальных изображениях.

Литература

[1] N. S. Guzhva, V. E. Prun, V. V. Postnikov, M. G. Lobanov, R. N. Sadekov and D. L. Sholomov, "Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene," *2022 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.

[2] N. S. Guzhva, B. Ali, K. S. Bakulev, R. N. Sadekov and A. V. Sholokhov, "Evaluating the Accuracy of Tram Positioning System in High-Rise Building Environment Using Data from Visual Geoinformation Systems," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-5, doi: 10.23919/ICINS51816.2023.10168407.

[3] K. Dergachov, S. Bahinskii and I. Piavka, "The Algorithm of UAV Automatic Landing System Using Computer Vision," *2020 IEEE 11th International* Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2020, pp. 247-252, doi: 10.1109/DESSERT50317.2020.9124998.

[4] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," *2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[5] D. V. Polevoy, A. Ingacheva, "From tomographic reconstruction to automatic text recognition: the next frontier task for the artificial intelligence"

[6] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale Video Classification with Convolutional Neural Networks. arXiv preprint arXiv:1412.0767. Available at: https://arxiv.org/abs/1412.0767

[7] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. arXiv preprint arXiv:1611.05431. Available at: https://arxiv.org/abs/1611.05431

[8] PyTorch Vision. (n.d.). DenseNet Implementation. Available at: https://github.com/pytorch/vision/blob/master/torchvision/models/densenet.py

[9] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. arXiv preprint arXiv:1611.05431. Available at: https://arxiv.org/pdf/1611.05431.pdf

[10] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. arXiv preprint arXiv:1411.4555. Available at: https://arxiv.org/pdf/1411.4555.pdf

[11] Дж. Смит и А. Джонсон, "Распознавание человеческих действий с использованием сверточных нейронных сетей", Журнал компьютерного зрения и обработки изображений, том 30, № 2, 2018, с. 45-62.

[12] Шапиро, Р. "Трансферное обучение в глубоком обучении: принципы и практика." https://arxiv.org/abs/1707.09725

[13] Шолле, Ф. "Xception: Deep Learning with Depthwise Separable Convolutions." https://arxiv.org/abs/1610.02357

[14] Huang, G. и др. "Densely Connected Convolutional Networks." https://arxiv.org/abs/1608.06993

[15] Kingma, D. P., и Ba, J. "Adam: A Method for Stochastic Optimization." https://arxiv.org/abs/1412.6980

[16] Методы оптимизации нейронных https://habr.com/ru/articles/318970/

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770-778) https://arxiv.org/abs/1512.03385

[18] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1800-1807). https://arxiv.org/abs/1512.03385

[19] 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 (CVPR) (pp. 4700-4708) https://arxiv.org/abs/1608.06993

[20] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980.

[21] https://www.researchgate.net/publication/367545589_A_Review_of_Navigation_Algorithms_for_Unmanned_Aerial_Vehicles_Based_on_Computer_Vision_Systems

[22] N. S. Guzhva, V. E. Prun, V. V. Postnikov, M. G. Lobanov, R. N. Sadekov and D. L. Sholomov, "Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene," 2022 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.