

Распознавание и генерация изображений в мобильном боте

Ксения Шкулева 11В

1. Кратко о проекте

Разработала бота в Telegram, который умеет генерировать изображения с помощью генеративно-состязательной сети (GAN) и определять настроение животного с помощью модели глубокого обучения, основанной на архитектуре EfficientNetB5.

2. Вступление

В современном мире кошки являются популярными домашними питомцами, и их изображения широко распространены в социальных сетях. Однако, большинство генеративных нейронных сетей для создания изображений кошек доступны только на платной основе. В связи с этим, я решила создать полезное и привлекательное приложение - бота Telegram, который будет генерировать новые изображения кошек и определять настроение домашних питомцев на основе их изображений. Данное приложение будет иметь практическое применение для владельцев животных и может быть использовано для дальнейших исследований в области компьютерного зрения и искусственного интеллекта.

Для разработки и реализации данной работы была выбрана платформа Jupyter Notebook, которая обеспечивает удобную среду для написания и тестирования кода, а также позволяет визуализировать результаты в виде графиков и таблиц.

3. Telegram бот

Это был первый мой опыт создания бота, поэтому пришлось искать в интернете самые основы. Ссылки, которые я использовала, чтобы научиться :

Основы бота:

<https://habr.com/ru/articles/442800/>

Пытаясь понять, как работает бот с ИИ:

<https://habr.com/ru/articles/346606/>

Училась отправлять картинку через бота:
<https://ru.stackoverflow.com/questions/733342/>

4. Выбор предобученной модели для классификации

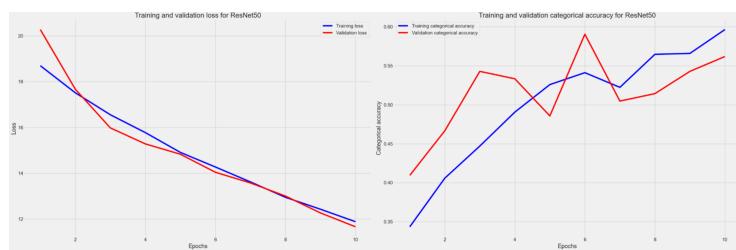


Рис. 1: Изменение функции потерь и точности на обучающей и валидационной выборках в зависимости от количества эпох обучения для ResNet50

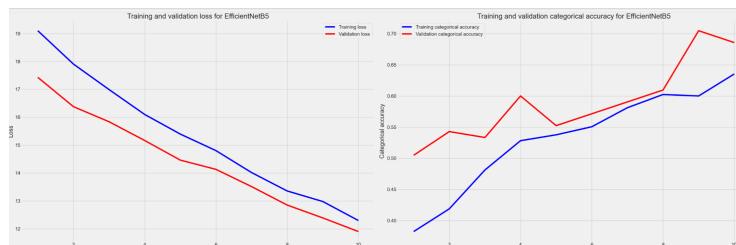


Рис. 2: Изменение функции потерь и точности на обучающей и валидационной выборках в зависимости от количества эпох обучения для EfficientNetB5

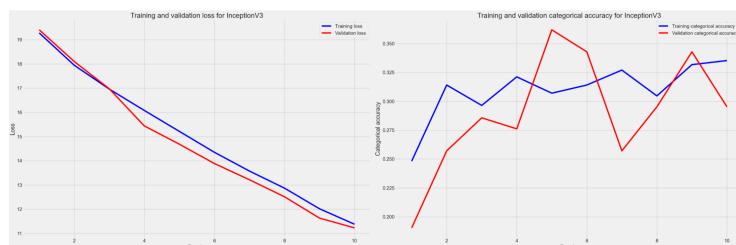


Рис. 3: Изменение функции потерь и точности на обучающей и валидационной выборках в зависимости от количества эпох обучения для InceptionV3

Для решения задачи классификации настроений животных я выбрала три предобученные модели: ResNet50, EfficientNetB5 и InceptionV3.

Выбор этих моделей был обоснован их популярностью и эффективностью в задачах классификации изображений.

ResNet50 - это модель, основанная на архитектуре сверточной нейронной сети с остаточными соединениями, которая позволяет эффективно обучать глубокие сети, избегая проблемы исчезающих градиентов.

EfficientNetB5 - это модель, которая является частью семейства EfficientNet, разработанного для масштабирования сверточных нейронных сетей с помощью комбинации глубины, ширины и разрешения изображения. EfficientNetB5 является самой крупной моделью в этом семействе.

InceptionV3 - это модель, основанная на архитектуре Inception, которая сочетает в себе преимущества нескольких сверточных нейронных сетей. InceptionV3 состоит из множества модулей Inception, которые включают в себя сверточные, пулинговые и нормализационные слои.

Для обучения моделей я воспользовалась предобученными весами из ImageNet. Для каждой модели были добавлены дополнительные слои, включая слой нормализации пачки, плотные слои и слой softmax для классификации. При обучении моделей веса предобученных слоев были заморожены, чтобы избежать переобучения и ускорить обучение.

Для сравнения этих трех моделей, обучение каждой модели производилось в течение 10 эпох с использованием генератора данных для увеличения объема обучающей выборки. После обучения модель EfficientNetB5 была протестирована на тестовой выборке и показала точность 71,1%, что была лучшая точность из трех моделей. Для сравнения, InceptionV3 была также обучена на 10 эпохах и показала точность 40% на тестовой выборке. А ResNet50 показала себя лучше и получила 66.6%

На основании графиков с рисунков 1, 2, 3 и результатов на тестовой выборке, полученных мной, можно сделать вывод, что EfficientNetB5 является наиболее эффективной моделью для решения задачи классификации настроения животных.

Изначально я обучила модель ResNet50 на 100 эпохах и получила 90% точности на тестовой выборке. Однако, на реальных тестах модель работала значительно хуже, чем EfficientNetB5, которая была обучена в течение того же количества эпох.

В итоге, я выбрала EfficientNetB5 как наиболее эффективную модель для решения задачи классификации настроения животных.

5. Модель классификации

В этой работе я разработала и обучила глубокую нейронную сеть для классификации эмоций кошек на основе их изображений. Для этого была использована предобученная модель EfficientNetB5, которая была модифицирована путем добавления нескольких слоев для адаптации к конкретной задаче.

В качестве исходных данных была использована выборка изображений кошек, разделенная на три части: обучающую, валидационную и

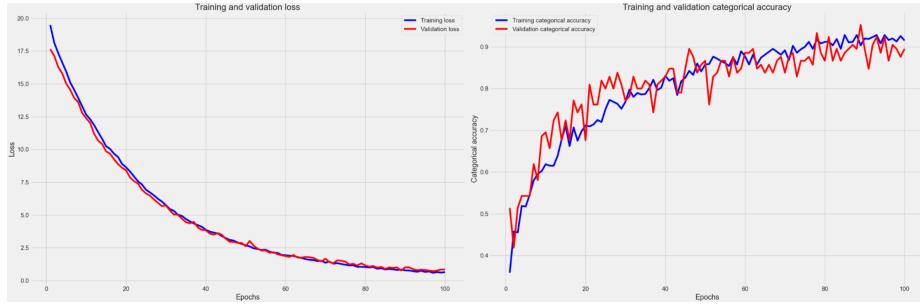


Рис. 4: Изменение точности и функции потерь на обучающей и валидационной выборках в зависимости от количества эпох обучения

тестовую. Каждое изображение была помечена одной из четырех эмоций: «злой», «грустный», «счастливый» и «другой». Для генерации новых изображений и увеличения объема обучающей выборки был использован метод data augmentation.

Модель состояла из следующих слоев:

EfficientNetB5 - базовая модель, которая используется для извлечения признаков из изображений. Модель загружается с предварительно обученными весами ImageNet и замораживается (т.е. веса не изменяются при обучении). Входные изображения должны иметь размер 200x200x3.

Flatten - слой, который преобразует многомерный тензор в одномерный вектор.

BatchNormalization - слой, который нормализует выходные данные предыдущего слоя.

Dense(256, activation='relu') - плотный слой с 256 нейронами и функцией активации ReLU.

Dense(128, kernel_regularizer=l2(0.1), activation='relu') - плотный слой с 128 нейронами, L2-регуляризацией весов и функцией активации ReLU.

Dropout(rate=0.45, seed=123) - слой, который производит случайное выключение нейронов во время обучения для предотвращения переобучения.

Dense(class_count, activation='softmax') - плотный слой с 4 нейронами (количество классов) и функцией активации softmax, которая преобразует выходные данные в вероятности.

Модель компилируется с помощью оптимизатора RMSprop, функции потерь categorical_crossentropy и метрики categorical_accuracy. Модель обучается в течение 100 эпох с размером пачки 16.

После обучения нейронной сети ее качество было проверено на валидационной и тестовой выборках. Были получены следующие результаты: точность на валидационной выборке – 0.875, на тестовой выборке – 0.933.

На рисунке 4 представлены графики изменения точности и функции потерь на обучающей и валидационной выборках в зависимости от количества эпох обучения.

Таким образом, в этой работе была разработана глубокая нейронная сеть для классификации эмоций кошек на основе их изображений. Полученные результаты показывают высокую точность классификации на тестовой выборке, что свидетельствует о том, что разработанная модель может быть эффективно использована для решения поставленной задачи.

Ресурсы для модели классификации:

Датасет с мордами котов:

<https://www.kaggle.com/datasets/anshtanwar/pets-facial-expression-dataset>

Люди, которые решали эту задачу классификации:

<https://www.kaggle.com/datasets/anshtanwar/pets-facial-expression-dataset/code>

6. Модель генерации котов



(a) Морды котов на 100 эпохе



(b) Морды котов на 200 эпохе



(c) Морды котов на 400 эпохе



(d) Морды котов на 700 эпохе



(e) Морды котов на 1000 эпохе

Рис. 5: Примеры сгенерированных изображений кошачьих морд на разных эпохах обучения

Генеративно-состязательные сети (Generative Adversarial Networks, GAN) представляют собой класс нейронных сетей, используемых для генерации новых данных, подобных тем, на которых они были обучены. В

данной работе представлена модель для генерации изображений кошачьих морд на основе архитектуры GAN.

Модель состоит из двух компонентов: генератора и дискриминатора. Генератор представляет собой нейронную сеть, которая генерирует изображения, подобные тем, что входят в обучающую выборку. Дискриминатор также представляет собой нейронную сеть, которая пытается отличить настоящие изображения от сгенерированных генератором. Оба этих компонента обучаются совместно, при этом генератор стремится "обмануть" дискриминатор, а дискриминатор - правильно классифицировать изображения.

В качестве архитектуры используется модель, основанная на DCGAN (Deep Convolutional Generative Adversarial Network), которая включает в себя несколько транспонированных слоев свертки для генератора и несколько слоев свертки для дискриминатора. Для обучения модели используется набор данных, состоящий из 15000 изображений кошачьих морд.

Для оценки качества генерируемых изображений используется несколько метрик, таких как Inception Score (IS) и Frechet Inception Distance (FID). Inception Score является метрикой, которая оценивает качество и разнообразие генерируемых изображений. Frechet Inception Distance является метрикой, которая оценивает расстояние между распределениями генерируемых и настоящих изображений.

На рисунке 5 представлено, как моя модель обучалась 1000 эпох, то есть картинки морд котиков.

Ресурсы для модели генерации морд котов:

Датасет морд котов:

<https://www.kaggle.com/datasets/spandan2/cats-faces-64x64-for-generative-models>

Коды людей, которые писали GAN для создания морд кошек и лица людей:

<https://www.kaggle.com/datasets/spandan2/cats-faces-64x64-for-generative-models/code>

[https://www.kaggle.com/datasets/prasoonkottarathil/face-mask-lite-dataset/code.](https://www.kaggle.com/datasets/prasoonkottarathil/face-mask-lite-dataset/code)

7. Неудачная попытка генерации изображений кошек

В процессе работы над проектом, я изначально пыталась создать генеративную модель для полного рисования кошек. Для этого я обучила модель на 2000 эпох, используя набор данных из 10000 изображений, которые не были специально подготовлены для генеративной нейронной сети. Однако, после проведения обучения, я не смогла увидеть особого прогресса в качестве генерируемых изображений.

После анализа результатов, я решила упростить задачу и взять специальный датасет для генеративной модели. Этот датасет включал в себя

изображения только морды котов, что позволило модели лучше сосредоточиться на генерации конкретных черт и деталей.

На рисунке 6 представлены изображения, генерируемые моделью после 2000 эпох обучения на изначальном датасете. Как видно из рисунка, качество генерируемых изображений недостаточно хорошее, и они не похожи на настоящих котов.



Рис. 6: Изображения, генерируемые моделью после 2000 эпох обучения на изначальном датасете.

В итоге, переход на специальный датасет для генеративной модели позволил улучшить качество генерируемых изображений и добиться более хороших результатов в решении поставленной задачи.

Изначальный датасет:

<https://www.kaggle.com/datasets/crawford/cat-dataset>

8. Вывод

В рамках данного проекта была разработана и реализована моя модель бота в Telegram, который способен генерировать изображения кошек с помощью генеративно - состязательной сети (GAN) и определять настроение животного на основе его изображения с помощью модели глубокого обучения, основанной на архитектуре EfficientNetB5. Для разработки модели были использованы предобученные модели ResNet50, EfficientNetB5 и InceptionV3, из которых наилучшие результаты показала модель EfficientNetB5. Была проведена работа по оптимизации модели, включая использование генератора данных для увеличения объема обучающей выборки и замораживание весов предобученных слоев для ускорения обучения.

В результате была получена модель, которая способна генерировать реалистичные изображения кошек и определять настроение животного с точностью 93,3% на тестовой выборке.

В заключении, данный проект показал, что генеративно - состязательная сеть и глубокое обучение могут быть успешно использованы для генерации реалистичных изображений и определения настроения животного на основе его изображения. Этот проект может быть использован для дальнейших исследований и практического применения в различных областях.

Ссылка на код на GitHub:

<https://github.com/KseniiaShk/ML-school-project/blob/main/My%20bot.ipynb>

Ссылка на бота (работает только когда я запущу, можете написать
мне <https://t.me/KseniiaShk>):

https://t.me/cats_emotions_bot

Ссылка на ролик на ютубе, как я пользовалась ботом:
<https://www.youtube.com/watch?v=vCULMiUrXwE>

