Цель работы

Работа представляет собой комплексное и многогранное исследование, которое включает в себя понимание архитектуры, обучение и применение этих сетей.

Задание

1. Подготовка данных
2. Обучение моделей, реализованных по 3 различным архитектурам (AlexNet, VGG16, VGG19)
3. Оценка результатов обучения модели

Выполнение работы

Разрабатывается нейронная сеть для распознавания на изображениях объектов 4 классов: «Здание», «Лес», «Водоем», «Гора». Гарантируется, что на одном изображении нет объектов, принадлежащих разным классам. Количество изображений в обучающей выборке для каждого класса – около 1050. В работе применяется фреймворк TensorFlow + Keras API.

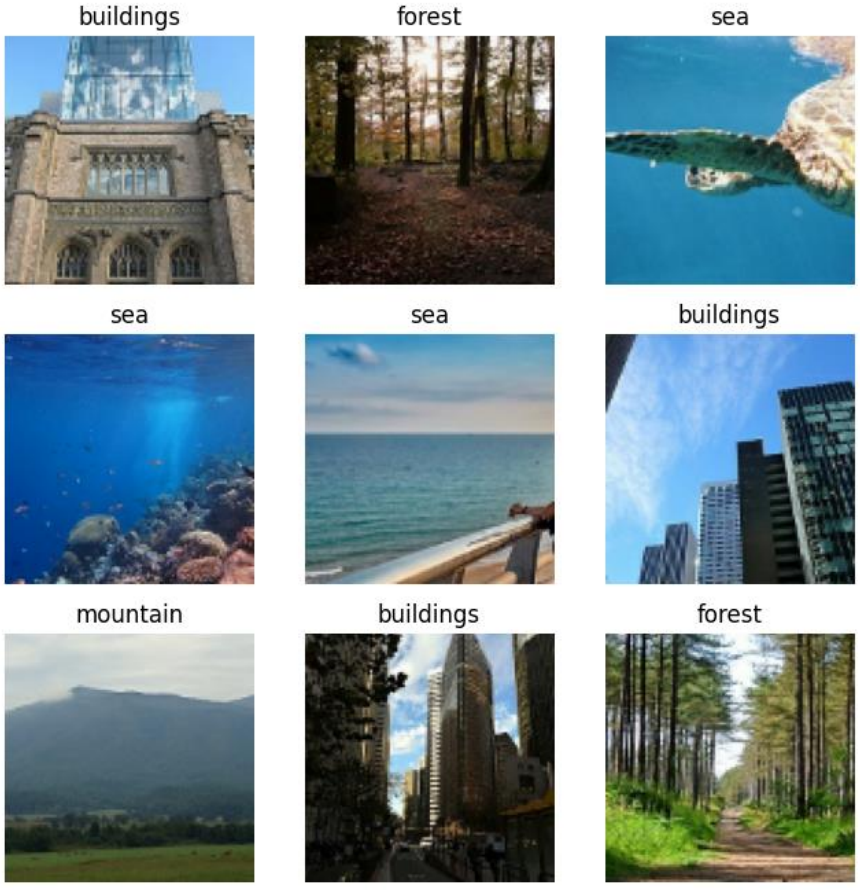


Рисунок 1. Примеры изображений каждого класса

Подготовка данных и разбиение на тренировочную, валидационную и тестовую выборку приведена на рис.2.

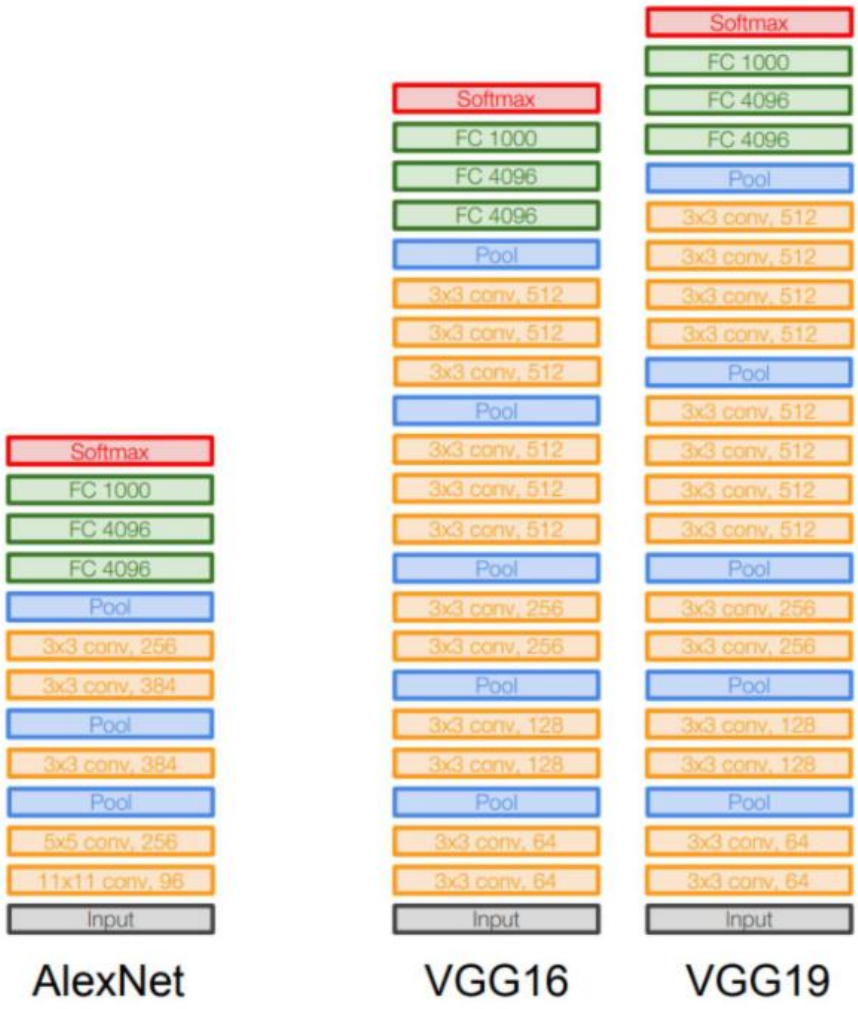
```
Found 4222 files belonging to 4 classes.
Using 3800 files for training.

Found 4222 files belonging to 4 classes.
Using 422 files for validation.

Found 1946 files belonging to 4 classes.
```

Рисунок 2

Затем были построены модели по архитектурам AlexNet, VGG16 и VGG19. Используемая функция активации – ReLU. Архитектуры состоят из сверточных, пулинговых и полносвязных слоев.



Ниже приведены результаты обучения (точность и функция потерь).

Архитектура AlexNet (14 эпох). Количество эпох было сокращено (early stopping) во избежание переобучения. Точность в конце 14 эпохи: accuracy: 0.6871 - val_accuracy: 0.6919

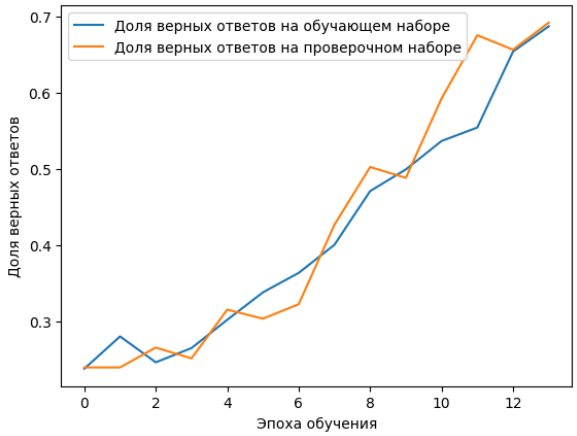


Рисунок 3

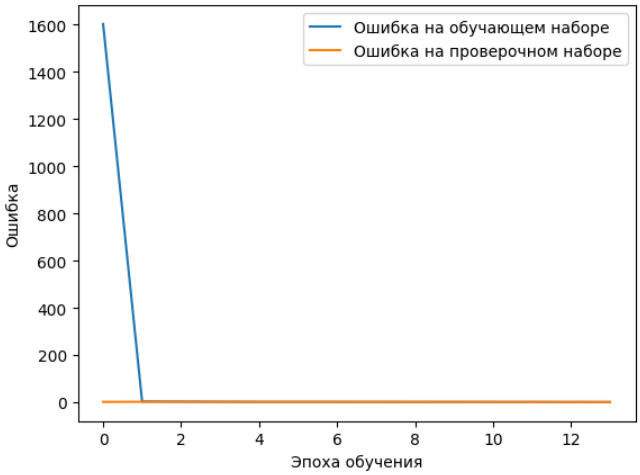


Рисунок 4

Архитектура VGG16 (25 эпох). Точность в конце 25 эпохи: accuracy: 0.8863 - val_accuracy: 0.8175

Рисунок 5

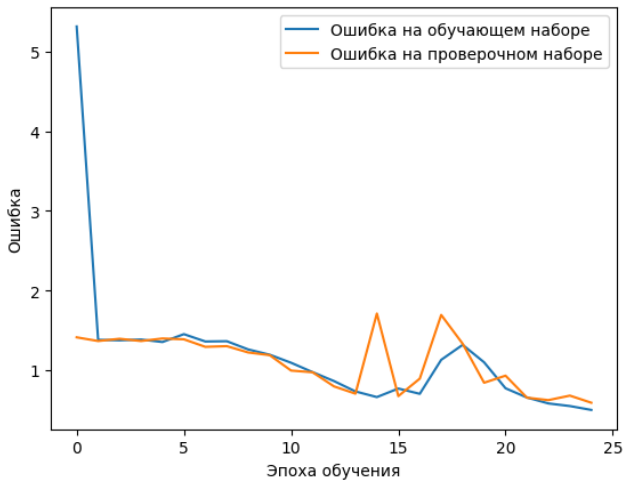


Рисунок 6

Архитектура VGG19 (25 эпох). Точность в конце 25 эпохи: accuracy: 0.6947 - val_accuracy: 0.7038

Рисунок 7

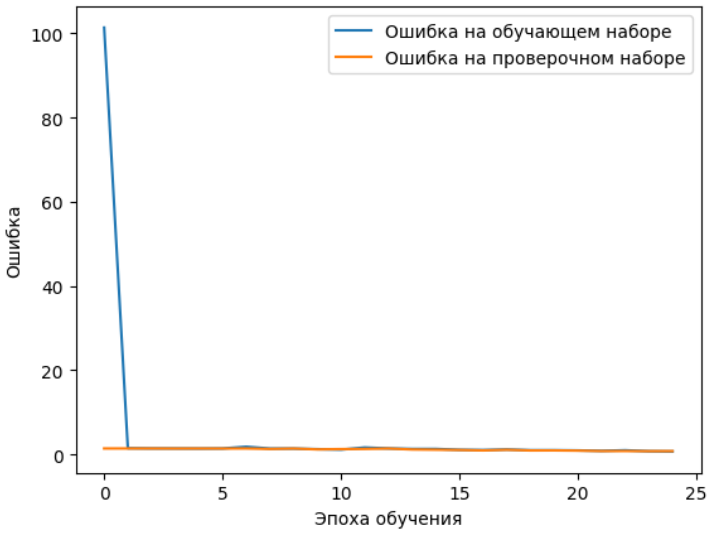


Рисунок 8

Каждая из представленных архитектур позволила достаточно хорошо обучить нейронную сеть, наибольшая точность (0.8863) была достигнута при использовании VGG16.

	Accuracy	Val. accuracy
AlexNet	0.6871	0.6919
VGG16	0.8863	0.8175
VGG19	0.6947	0.7038

Для избежания переобучения на моделях была использована функция Dropout(). Модели, реализованные разными архитектурами, обучались различное количество эпох. Оптимальное количество эпох — это область между состояниями недообучения и переобучения модели. Пример переобучения на модели с архитектурой VGG16 (50 эпох), приведен на рис.9.

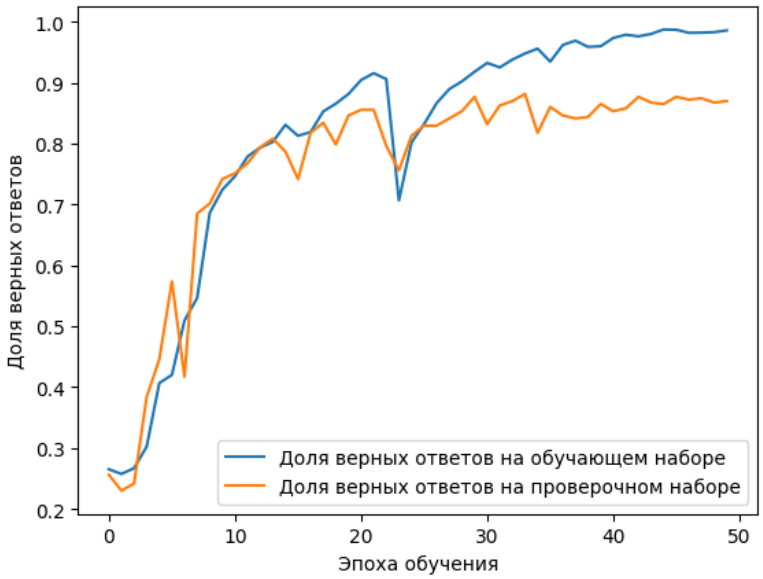


Рисунок 9

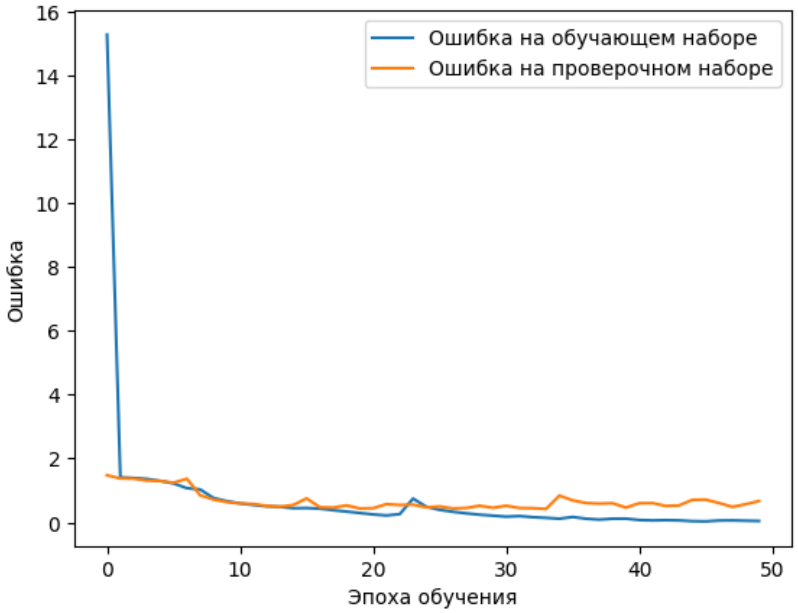


Рисунок 10

По рисунку 9 видно, что после 25-30 эпох модель начинает переобучаться, поскольку метрика на валидационном наборе данных перестает улучшаться. Также это можно увидеть по графику на рис.10 – после 25 эпох функция потерь на тестовом наборе стала превышать значение функции на обучающем наборе.

Ссылки на Google Colab:

AlexNet: https://colab.research.google.com/drive/1RMpOgPbMz28EUAaVojcEcl-TvvxhE8N6?usp=sharing

VGG16: https://colab.research.google.com/drive/1HsO52meZlmlIeDn4MDQbhanbgSbA9Gkg?usp=drive_link

VGG19: https://colab.research.google.com/drive/1CCyxZq6g3cglKB5--I4msUWb_9DgGwGM?usp=drive_link

VGG16(overfitting): https://colab.research.google.com/drive/1a9xd7iF1EEemfoNn632TkJX8rIKaSSST?usp=sharing

Выводы

В результате выполнения работы было проведено исследование сверточных нейронных сетей, включающее в себя подготовку и разметку данных, разработку архитектуры и обучение моделей по 3 различным архитектурам, а также оценку результатов обучения модели. Проведено сравнение точности моделей, наилучшей оказалась архитектура VGG16. Проведена регуляризация (использование dropout), а также рассмотрено явление переобучения.