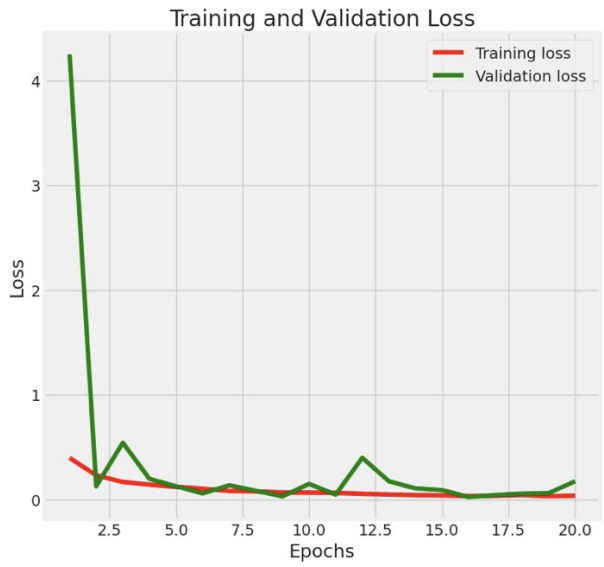
На первом шаге мы импортируем необходимые для работы библиотеки.

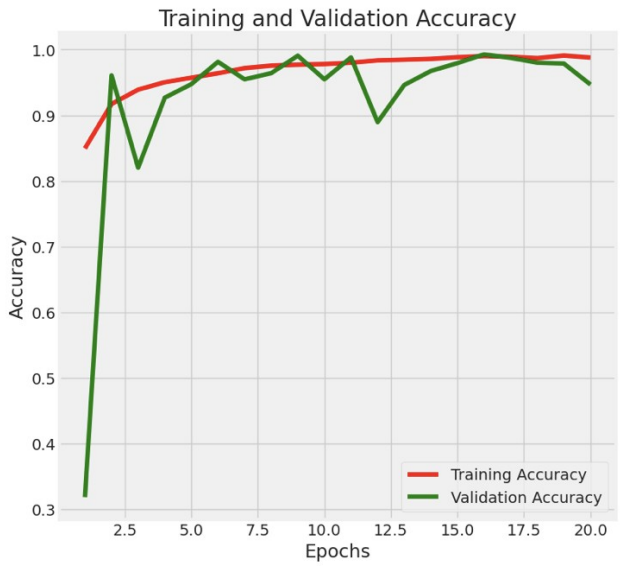
Далее подгружаем датасет с гистологическими изображениями рака легкого, оформляя его в ДатаФрейм, где в первом столбце фиксируется путь до изображения, а во втором - название (класс). Получилось 15000 строк и 2 столбца.

Проводим предобработку данных. Построим круговую диаграмму, которая поможет увидеть распределение групп по частоте встречаемости. В нашем случае выборка балансирована - имеет три группы, каждая составляет 33.3%
Далее делим данные на обучающий (80% - 12000 изображений), валидационный (10% - 1500 изображений) и тестовый (10% - 1500 изображений) наборы.
Нормализуем все изображения, деля их пиксели на 255 и, таким образом, получаем на выходе размер каждого пикселя в диапазоне от 0 до 1.
Выводим батч размером 32 элемента с указанными классами.

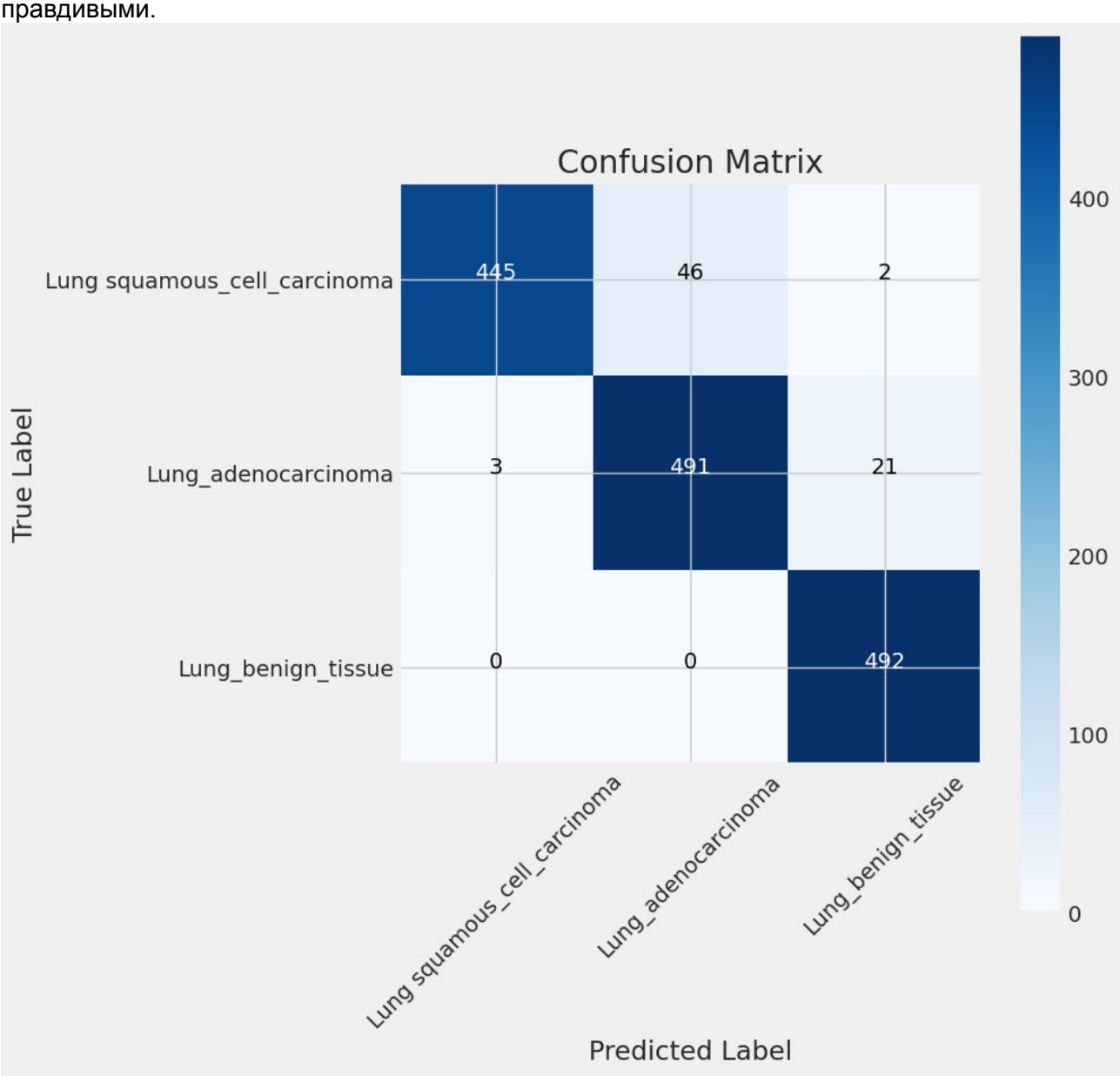
Создаем несколько функций - первая (model_performance) выводит два графика, где сравниваются данные по значению точности и значению функции потерь на тренировочной и валидационной выборках.
Вторая функция (model_evaluation) будет оценивать модель на трех группах данных с указанием точности и потерь.
Третья (get_pred) - выводит индекс максимального значения в массиве
Четвертая (plot_confusion_matrix) - выводит матрицу ошибок в виде графика
Пятая (conv_block) - определяет сверточный блок для CNN
Шестая (dense_block) - определяет полносвязный блок для CNN

Создаем структуру модели сверточной нейронной сети (CNN). Проводим компиляцию модели. Далее обучаем модель. Количество эпох равно 20. Время, заточенное на каждой эпохе составляет в среднем 60 с. Мы наблюдаем увеличение точности на обучающих данных с 0.7749 до 0.9878, и на валидационных данных - с 0.3180 до 0.9467. Значения функции потерь же наоборот снижаются на обучающих данных с 0.5612 до 0.0367, и на валидационных - с 4.2505 до 0.1745. На графиках оценки модели мы видим тот же результат.



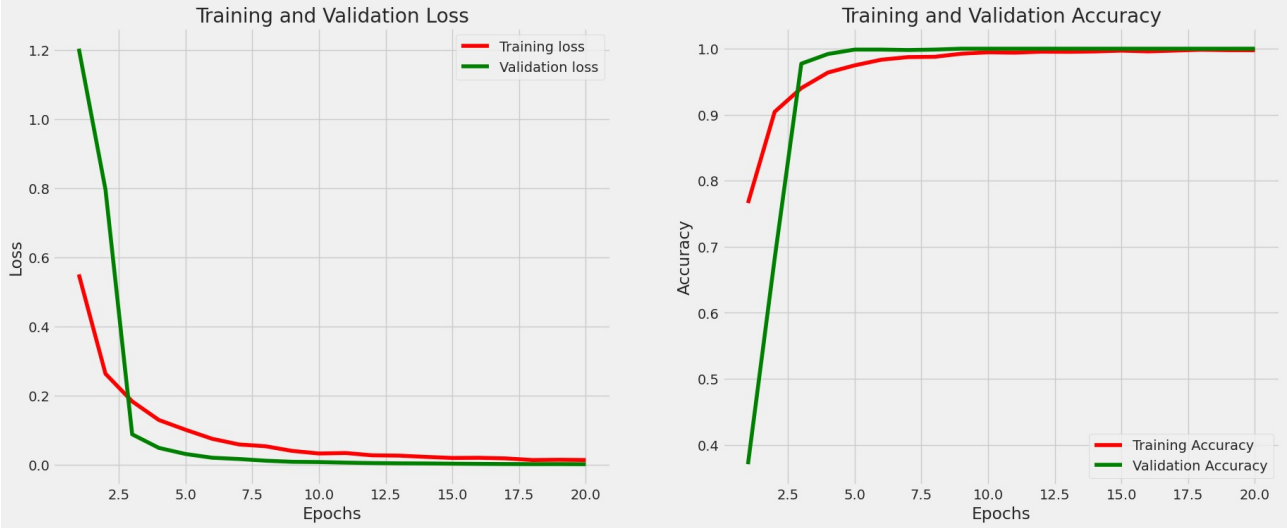


Результаты матрицы ошибок указывают на довольно хорошую работу модели - предсказанные по изображениям названия классов в большинстве случаев совпадают с правдивыми.



Создаем еще одну модель - EfficientNetB3. Загружаем ее с предобученными весами на ImageNet, исключаем верхний слой, так как будем добавлять свой собственный. Уменьшаем размеренность признаков и добавляем слои. Проводим компиляцию.

Обучаем модель (также 20 эпох). Видим, что работает она дольше (в среднем 100 секунд за 1 проход). Точность на обучающей выборке растет с 0.6579 до 0.9983, на валидационной - с 0.3707 до 1.0000. А функция потерь на обучающих данных падает с 0.7873 до 0.0126, а на валидационных - с 1.2038 до 0.0018. Можем отметить, что данная модель имеет более совершенные характеристики и это же подтверждает график.



Финальная оценка показывает, что точность на обучающей и валидационной выборках составляет 1.0, на тестовой - 0.999. Функции потерь на обучающий 0.01, на валидационной 0.02, она тестовой 0.03

Матрица ошибок демонстрирует более точное предсказание по сравнению с предыдущей моделью.