Московский государственный университет имени М.В.Ломоносова
Факультет вычислительной математики и кибернетики

Отчет о выполнении практической работы «Метрические алгоритмы классификации»

Петрова Александра,
317 группа

Москва
2023

Оглавление

Введение

Данное практическое задание посвящено исследованию метрических алгоритмов классификации на примере обучения моделей k ближайших соседей для решения задачи распознавания цифр с изображений из датасета MNIST.

Целью исследования является выявление зависимости точности (доли верно предсказанных ответов) от числа ближайших соседей k, метрики, способа аугментации.

Пояснения к задаче

Для проведения экспериментов были написаны реализация методов поиска ближайших соседей и кросс-валидации на языке `Python`.

Рассматривались следующие алгоритмы поиска ближайших соседей:

- `'my_own'` — собственная реализация (использовалась функция `numpy.argpartition`)
- `'brute'` — использование `sklearn.neighbors.NearestNeighbors(algorithm='brute')`
- `'kd_tree'` — использование `sklearn.neighbors.NearestNeighbors(algorithm='kd_tree')`
- `'ball_tree'` — использование `sklearn.neighbors.NearestNeighbors(algorithm='ball_tree')`

Датасет MNIST состоит из 70000 изображений. В качестве обучающей выборки были выбраны первые 60000 изображений, в качестве тестовой - оставшиеся.

Эксперименты

Эксперимент №1

В данном эксперименте измерялось время работы различных алгоритмов поиска ближайших соседей в зависимости от размера признакового пространства. В качестве метрики бралось евклидово расстоение. Результаты эксперимента представлены на Рис.1.

Резкий рост времени работы алгоритмов 'kd_tree' и 'ball_tree' с увеличением размерности признакового пространства объясняется тем, что при размерностях больших 20 сложность поиска ближайшего соседа в дереве сильно ухудшается и приобретает линейный порядок сложности.

Алгоритм 'brute' работает быстро, потому что не требует предварительной обработки данных или построения дополнительных структур данных.

Эксперимент №2

Во втором эксперименте находилась зависимость точности алгоритмов от количества ближайших соседей и метрики. Так как все алгоритмы, которые участвуют в экспериментах, являются точными методами нахождения ближайших соседей, достаточно выбрать один из них, самый быстрый. По результатам первого эксперимента был выбран алгоритм 'brute'. Точность оценивалась с помощью кросс-валидации с 3 фолдами. Результаты проиллюстрированы на Рис.2. По рисунку видно, что косинусная метрика при любом k даёт более точный результат, чем евклидова. Выброс при $k = 2$ объясняется тем, что вероятность ошибиться при выборе одного из двух разных ответов равна 0.5. При росте k точность понижается, потому что в соседей попадают объекты из других классов.

Также в рамках этого эксперимента было посчитано время нахождения трех ближайших соседей для алгоритмов 'brute' и 'my_own' в зависимости от метрики. Эти алгоритмы были

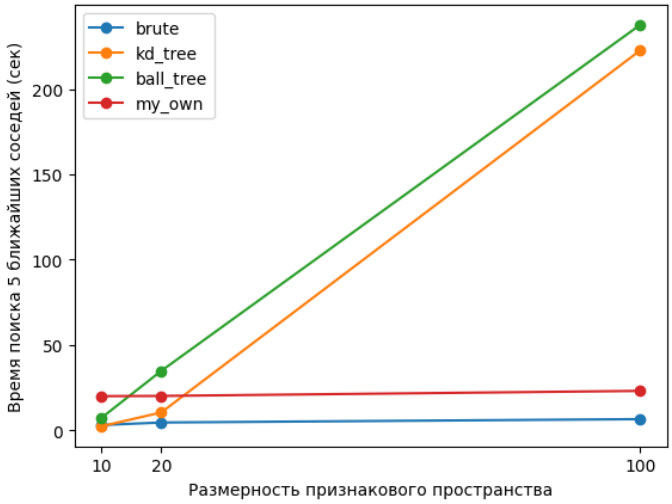

Рис. 1: График зависимости времени работы алгоритмов от размера признакового пространства

Алгоритм	Косинусная метрика	Евклидова метрика
brute	175	131
my_own	166	170

Таблица 1: Время поиска 3 ближайших соседей в зависимости от алгоритма и метрики, в секундах

выбраны, потому что стратегии 'kd_tree' и 'ball_tree' не могут принимать в качестве параметра косинусную метрику. Результаты показаны в Таблице 1.

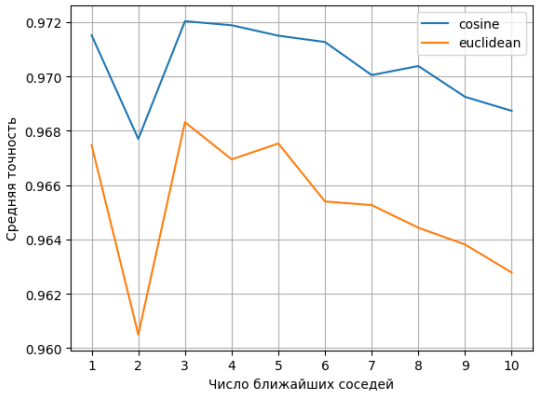

Рис. 2: График зависимости средней точности по кросс-валидации от k

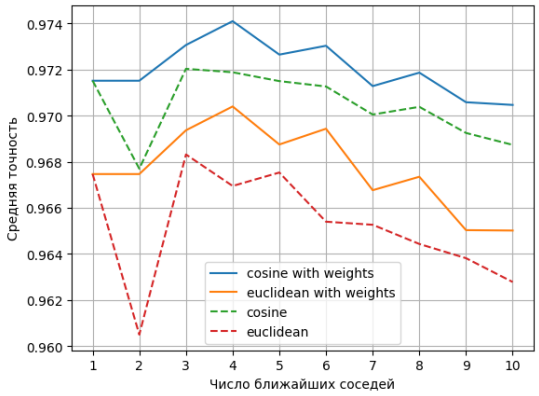

Рис. 3: График зависимости средней точности по кросс-валидации от k для взвешенных и не взвешенных алгоритмов

Эксперимент №3

В третьем эксперименте сравнивался взвешенный метод k ближайших соседей с методом без весов при тех же фолдах и параметрах. Веса объектов устанавливались в соответствии с формулой $1/(distance + \varepsilon)$, где $\varepsilon - 10^{-5}$. Средняя точность по 3 фолдам в зависимости от метрики и k изображена на Рис.3, время поиска ближайших соседей при $k = 3$ в зависимости от метрики отображено в Таблице 2.

Как видно из Рис.3, взвешенный метод точнее соответствующего невзвешенного. Взвешенный алгоритм устраняет выброс при k = 2.

Таким образом, взвешенный метод k ближайших соседей учитывает значимость каждого из ближайших соседей и может давать более точные прогнозы, чем невзвешенный метод, который не учитывает этот фактор.

Эксперимент №4

По результатам предыдущих экспериментов был выбран лучший алгоритм поиска k ближайших соседей: взвешенный, стратегия - brute, число ближайших соседей - 4, метрика - косинусная. Точность алгоритма на тестовой выборке составила 0.9752, по кросс-валидации с 3 фолдами - 0.9731. Точность лучших алгоритмов на данной выборке более 0.997 (`https://www.kaggle.com/code/cdeotte/mnist-perfect-100-using-knn#Result-100%-classification-accuracy-!!`)

По предсказанным и реальным таргетам была построена матрица ошибок, приведенная на Рис.16. Больше всего модель ошибается на парах (сначала реальный, потом предсказанный): 4 - 9, 7 - 9, 3 - 5, 7 - 1. На Рис.4 - Рис.7 показаны некоторые изображений, на которых происходят ошибочные предсказания.

Модель ошибается в случаях, когда цифры повернуты, сдвинуты относительно центрального положения, имеют нечёткие границы.

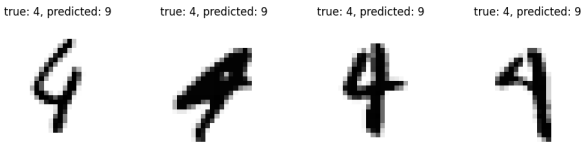

Рис. 4: 4 распознана как 9

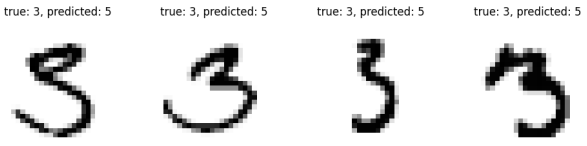

Рис. 5: 3 распознана как 5

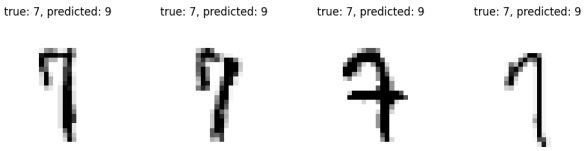

Рис. 6: 7 распознана как 9

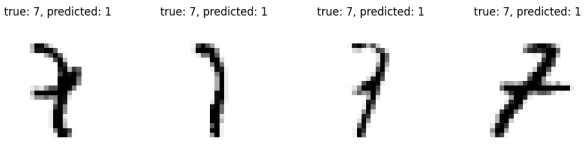

Рис. 7: 7 распознана как 1

Эксперимент №5

В этом разделе проводилась аугментация обучающей выборки. Были рассмотрены следующие преобразования:

1. Поворот. Величина поворота: 5, 10, 15 (в каждую из двух сторон).
2. Смещение. Величина смещения: 1, 2, 3 пикселя (по каждой из двух размерностей).
3. Гауссовский фильтр. Дисперсия фильтра Гаусса: 0.5, 1, 1.5.
4. Морфологические операции: эрозия, дилатация, открытие, закрытие с ядром 2.

В целях экономии времени перебирались не все возможные комбинации смещений. Выборка делилась на 4 части. Первая часть смещалась на s вверх и s влево, вторая - на s вверх и s вправо, третья - на s вниз и s влево, четвертая - на s вниз и s вправо, где s - величина смещения.

В начале	Поворот	Сдвиг	Фильтр Гаусса	Эрозия	Дилатация	Открытие	Закрытие
0.9476	0.9636	0.9533	0.952	0.9516	0.9496	0.949	0.9266

Таблица 2: Точность на объектах обучающей выборки

Параметры подбирались кросс-валидации с 3 фолдами. Преобразования применялись только к обучающей подвыборке во избежание переобучения. на валидационной выборке считалась точность и усреднялась по всем разбиениям. Находился максимум по параметрам преобразования. . Подобранные параметры:

1. Поворот: 5.

2. Смещение: 1.

3. Дисперсия фильтра Гаусса: 0.5.

Замечание: Для экономии времени перебор параметров при кросс-валидации производился не на всей обучающей выборке, а на её случайной подвыборке длины 18000. Таргеты имеют равномерное распределение, поэтому в случайной подвыборке всех ответов будет поровну. Перед тестированием каждого параметра выбиралась новая случайная подвыборка дл справедливости результатов. Точность на тестовой выборке всегда считалась на первых 3000 объектах (пропорционально доле всей тестовой от всей обучающей).

На Рис.8 - Рис.11 показано, как меняется матрица ошибок после каждого преобразования, примененного к оригинальной выборке. Также в Таблице 2 показаны точности после преобразований. Самые значимые изменения вносит поворот, потом сдвиг, потом фильтр Гаусса. Эти преобразования помогают уменьшить ошибки, связанные с поворотом, сдвигом и нечёткими границами цифр.

На оригинальной выборки были два лучших преобразования: поворот на 5 градусов и сдвиг на 1. Получена точность: 0.9788, что на 0,0036 больше точности без аугментации.

Эксперимент №6

В рамках данного эксперимента происходило обучение модели на оригинальной выборке, преобразование объектов тестовой выборки путем поворота, сдвига, Гауссовского фильтра, морфологических операций, применение модели к размноженной тестовой выборке и получение результата путем голосования среди преобразованных объектов.

В результате применения поворота размер тестовой выборки увеличивался в 3 раза и в 2 раза при остальных преобразованиях. Так как выбор из двух вариантов может приводить к случайному выбору, были проведены эксперименты увеличения выборки в 4 раз, но они не привели к значительным улучшениям.

В Таблице 3 показаны точности, полученные после каждого из преобразований по отдельности. Низкая точность у морфологических операций говорит о том, что в обучающей выборке не достаточно информации, чтобы распознавать морфологические изменения.

Путем перебора были выбраны параметры, на которых достигалась самая высокая точность. Они получились такими же как в эксперименте 5:

1. Поворот: 5.

2. Смещение: 1.

Матрицы ошибок для этих преобразований приведены на Рис.16 - Рис.23 Точности представлены в Таблице 3.

Первый помого исправить..., второй ..., третий ...

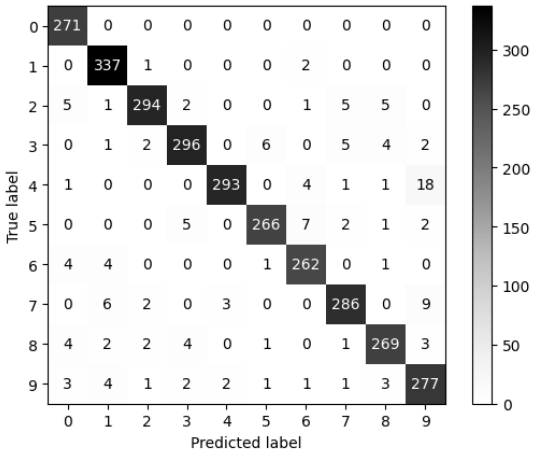

Рис. 8: Матрица ошмбок без аугментации

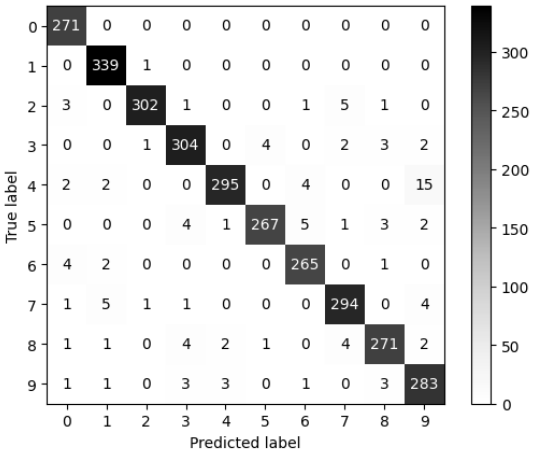

Рис. 9: Матрица ошибок с поворотом

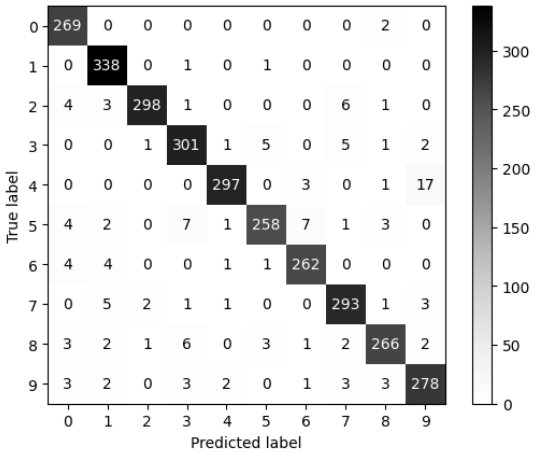

Рис. 10: Матрица ошибок со сдвигом

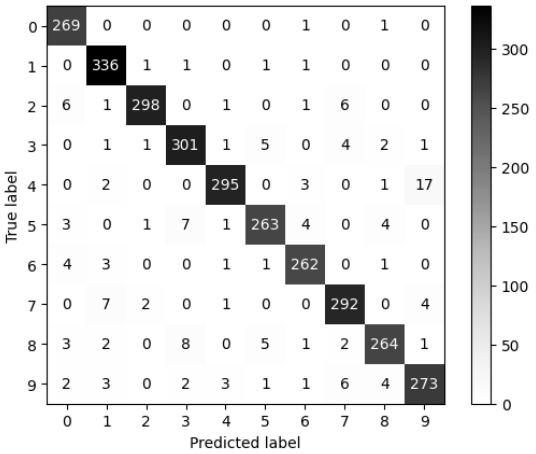

Рис. 11: Матрица ошибок с фильтром Гаусса

В начале	Поворот	Сдвиг	Фильтр Гаусса	Эрозия	Дилатация	Открытие	Закрытие
0.9752	0.9773	0.9433	0.9724	0.6731	0.7198	0.5459	0.5502

Таблица 3: Точность после голосования

К тестовой выборке была применена суперпозиция поворота на 5 градусов и сдвига на 1 пиксель. Получена точность: 0.9593 (на 0.0169 меньше точности без аугментации). Точность только при повороте составила 0.9773 (на 0.021 больше точности без аугментации)

Таким образом, первый подход показал результат на 0.015 точнее второго.

Качественно сравнивая два подхода к аугментации, стоит отметить, что второй более в первом случае модель получает большее количество информации, в результате чего может выдать более точный ответ. Во втором - точность может достигаться за счёт выбора наиболее популярного таргета для сильно похожих объектов. Однако, второй подход может не дать значимых улучшений, если в обучающей выборке недостаточно данных.

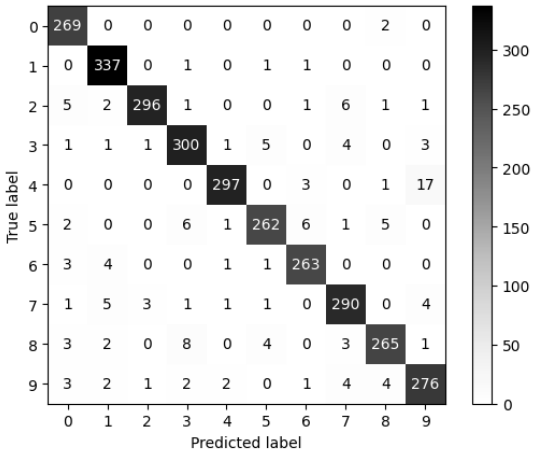

Рис. 12: Матрица ошибок с эрозией

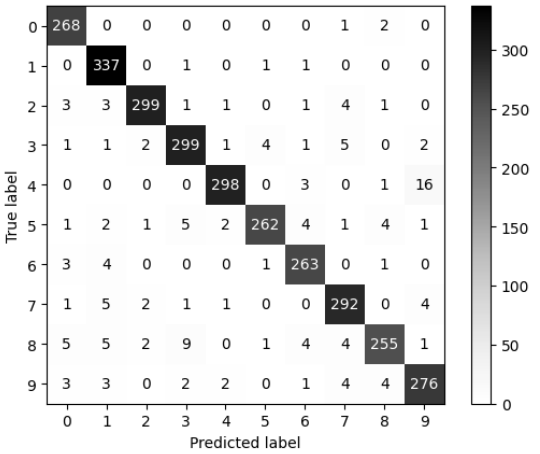

Рис. 13: Матрица ошибок с дилатацией

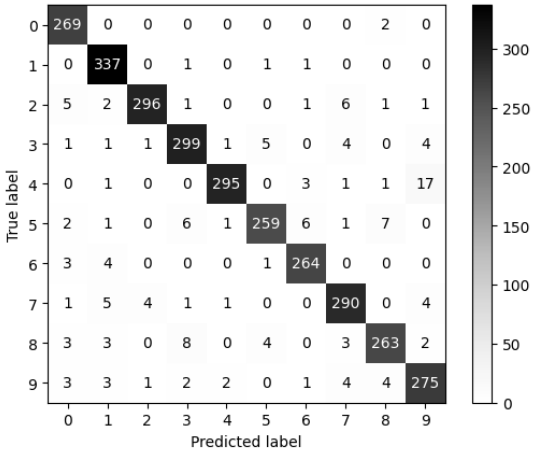

Рис. 14: Матрица ошибок с открытием

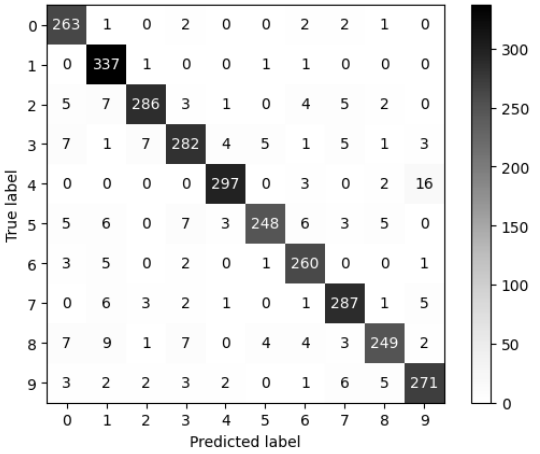

Рис. 15: Матрица ошибок с закрытием

Заключение

В данной работе были исследованы алгоритмы поиска k ближайших соседей в рамках решения задачи распознавания цифр с изображений из датасета MNIST.

В результате исследования выявлена зависимость точности от числа ближайших соседей k, метрики, взвешенности алгоритма и способа аугментации.

Лучшие результаты были получены при взвешенном алгоритме, $k = 4$ и косинусной метрике. Наилучшую точность 0.9788 удалось получить при помощи поворота и сдвига выборки.

Был произведен качественный анализ двух различных подходов аугментации данных - преобразования обучающей и тестовой выборок. Первый подход привел к более точным результатам на датасете MNIST.

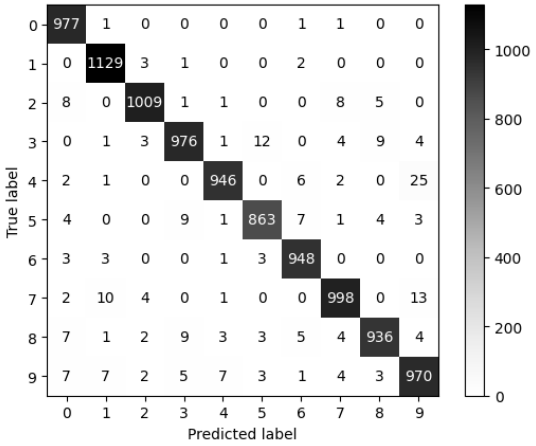

Рис. 16: Матрица ошибок без аугментации

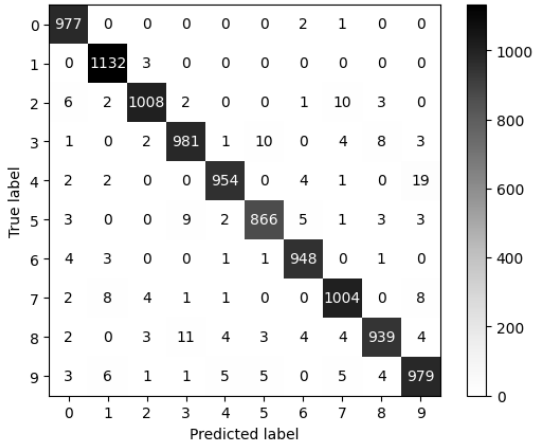

Рис. 17: Матрица ошибок с поворотом

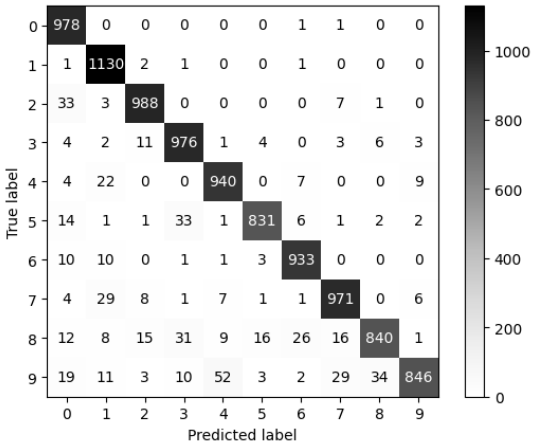

Рис. 18: Матрица ошибок со сдвигом

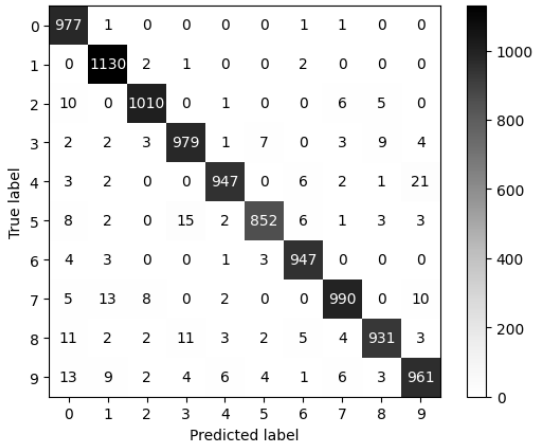

Рис. 19: Матрица ошибок с фильтром Гаусса

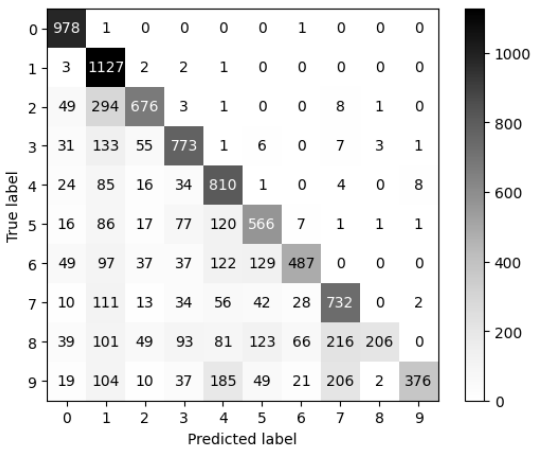

Рис. 20: Матрица ошибок с эрозией

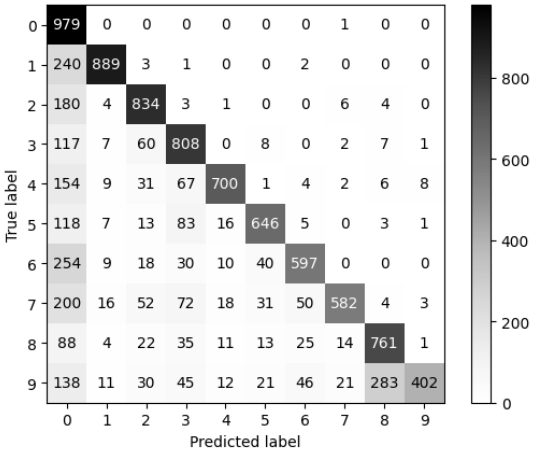

Рис. 21: Матрица ошибок с дилатацией

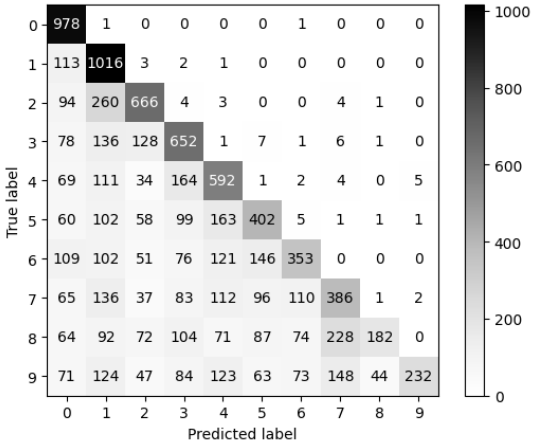

Рис. 22: Матрица ошибок с открытием

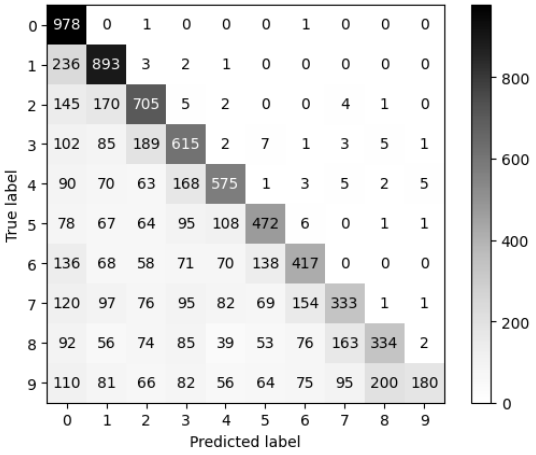

Рис. 23: Матрица ошибок с закрытием