Отчет по классификации открытых/закрытых глаз

Мельникова Елена

17 сентября 2024 г.

1 Постановка задачи

Требуется обучить классификатор открытых/закрытых глаз с использованием предоставленной обучающей выборки.

2 Кластеризация

Обучающая выборка не содержала меток классов открытых или закрытых глаз. Для решения данной задачи было принято решение применить кластеризацию для предварительной разметки данных.

Для извлечения признаков из изображений глаз были использованы `Dlib` и `VGG-Face`[1] . Модель `VGG-Face` была разработана специально для задач распознавания лиц и является адаптацией архитектуры VGG для работы с изображениями лиц. Она обучена на большом наборе данных лиц и демонстрирует отличные результаты при распознавании лиц. Библиотека `Dlib` содержит предобученную модель для распознавания лиц, которая использует ResNet архитектуру для извлечения признаков. Модель обучена на большом наборе данных лиц и может использоваться для задач идентификации и верификации.

Эти модели были выбраны, поскольку они обучены на изображениях лиц и, вероятно, могут эффективно извлекать признаки, связанные с глазами. Для оценки качества кластеризации была вычислена метрика `Silhouette_score`.

[1] https://github.com/serengil/deepface

Метод кластеризации	Dlib	VGG-Face
KMeans	0.0863	**0.1136**
AgglomerativeClustering	0.0621	0.0969
DBSCAN	0.0606	0.0356
BisectingKMeans	0.0873	**0.1093**

Таблица 1: Silhouette_score для разных кластеризаторов и фичей

Как видно из таблицы, алгоритмы, работающие с признаками, извлеченными с помощью модели VGG-Face, показали лучшие результаты. Наиболее эффективными оказались BisectingKMeans и KMeans.

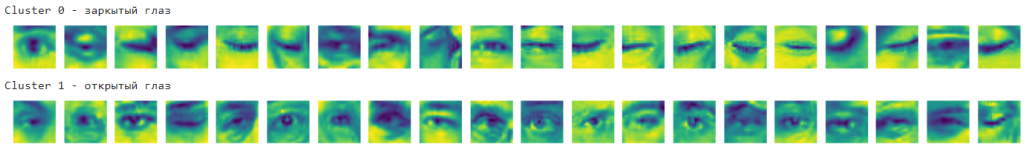


Рис. 1: Пример кластеризации KMeans на признаках от Dlib. Видно, что некоторые изображения были размечены некорректно, но в большинстве случаев результат правильный.

Для повышения качества кластеризации была использована библиотека Optuna для подбора гиперпараметров лучших моделей кластеризации BisectingKMeans и KMeans. Оптимизация позволила незначительно улучшить метрику.

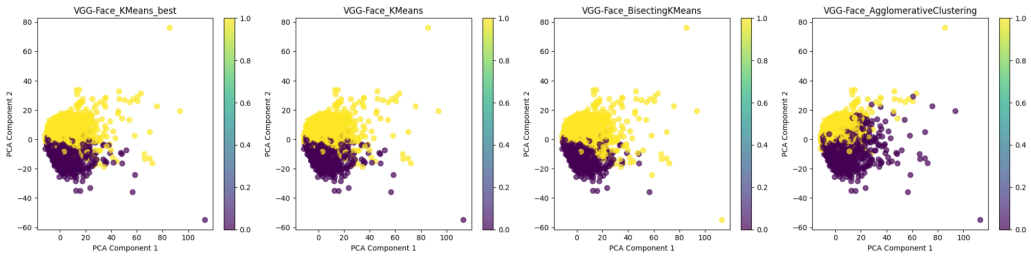


Рис. 2: Визуализация лучших алгоритмов кластеризации.

Для улучшения качества разметки датасета были выбрали семплы, по которым все хорошие модели (все, кроме DBSCAN на VGG-Face фичах) кластеризации дали одинаковый результат. После фильтрации датасета по согласованным результатам метрика улучшилась до 0.22, что в два

раза превышает показатель лучшей модели кластеризации с оптимизированными гиперпараметрами. Однако размер отфильтрованного датасета составил лишь четверть от исходного объема данных (около 1000 семплов).

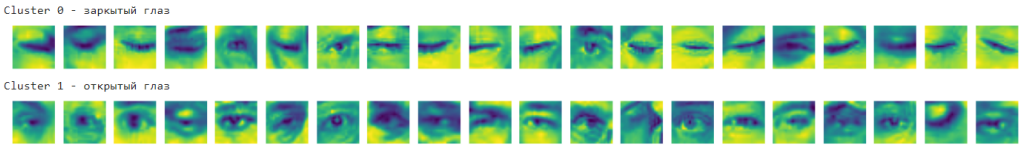


Рис. 3: Итоговое разбиение по согласованным результатам

Был также опробован выбор наиболее частого предсказания среди предсказаний всех моделей, однако такая кластеризация по метрике оказалась хуже, чем `KMeans` кластеризация.

3 Классификация

Для классификации в качестве бекбона использовалась модель `InceptionResnetV1`[2], предобученная на `vggface2`, так как она показала высокую эффективность в задачах классификации лиц. Решила не брать те же сети, что и для кластеризации, чтобы увеличить вероятность отловить ошибки в полученной разметке. Финальный слой был заменен на линейный с двумя выходными классами. Все веса модели были размороженны для обучения (Были проведены эксперементы с замороженными весами, но лучшие результаты удалось получить при полной разморозке). В качестве функции потерь использовалась кросс-энтропия (классы сбалансированы).

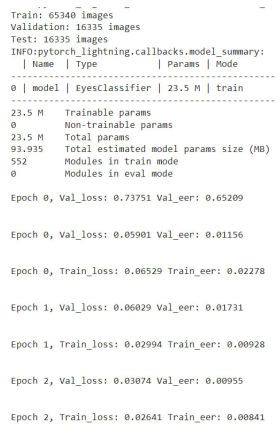


Рис. 4: Обучение на датасете с `Kaggle`

3.1 Обучение на внешнем датасете

Поскольку исходный датасет был небольшим и содержал ошибки в разметке, было решено использовать внешний размеченный датасет с `Kaggle`[3].

[2]https://github.com/timesler/facenet-pytorch
[3]https://www.kaggle.com/datasets/tauilabdelilah/mrl-eye-dataset

Модель обучалась на этом датасете в течение 10 эпох. Лучшая метрика `EER` и лосс была достигнута на 4 эпохе (< 0.01 на валидации), после чего модель начала переобучаться. Лучшую модель сохранила как `kaggle_model`.

3.2 Обучение на маленькой выборке

Затем `kaggle_model` была дообучена на выборке, полученной после кластеризации, состоящей из 1000 семплов, которые были разделены на тренировочную и тестовую подвыборки. Был проведён сравнительный анализ инференса на тестовой выборке, используя эту модель (`best_1000`), и инференса на той же тестовой выборке моделью `kaggle_model`. Даже обычная `kaggle_model` показывала хорошие результаты (EER $= 0.07$), однако допускала ошибки на изображениях с закрытыми глазами и длинными ресницами, классифицируя их как открытые. Вероятно, в открытом датасете таких семплов не было, что и привело к ошибкам. После дообучения эти ошибки исчезли.

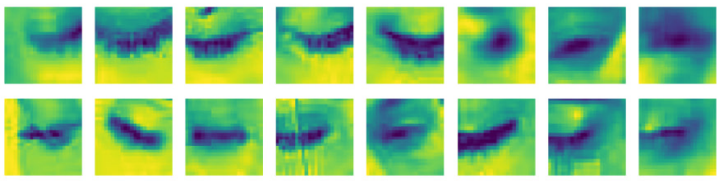
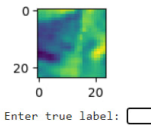


Рис. 5: Семплы где `kaggle_model` ошибочно приняла глаз за открытый

3.3 Обучение на большой выборке

Затем оставшаяся часть данной мне выборки, состоящая из 3000 семплов, была размечена с помощью `best_1000` - хорошей модели, судя по метрикам на предыдущем шаге.



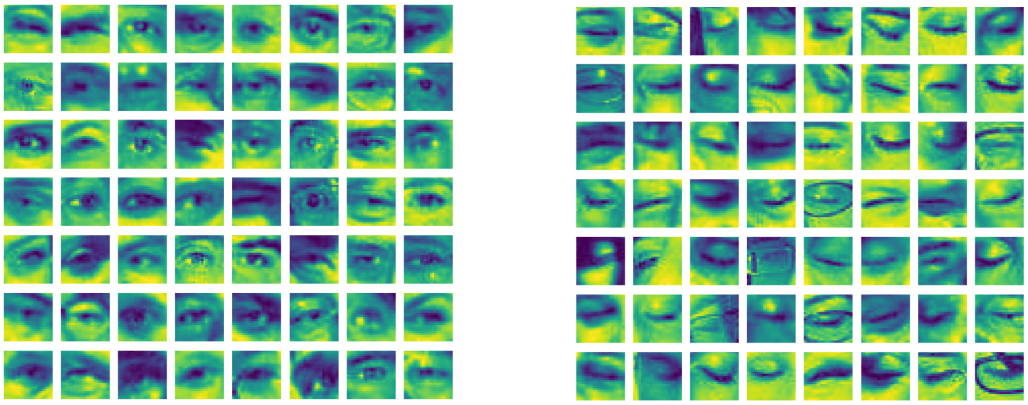
Изображения, для которых уверенность модели была меньше 0.05 и больше 0.95 были приняты за верные. Оставшиеся неуверенные семплы были размечены вручную (около 200). При ручной разметке стало ясно, что некоторые семплы сложно распознать даже человеку.

Получив хорошо размеченный датасет, он был разделен на трейн и тест выборку. Модель `kaggle_model` была снова дообучена, но теперь на большем датасете (из 3000 семплов). Модель `best_1000` больше не использовалась, так как с ее помощью были размечены данные. Получилась итоговая модель `final_best`. Метрика получились чуть хуже

чем на best_1000, однако и данные были сложнее - так как кластеризация не смогла с ними справиться. Инференс final_best на тестовой выборке для best_1000 показал примерно такие же результаты.

Модель	train_size	test_size	train_EER	val_EER	test_EER
kaggle_model	80.000	3000	0.002	0.01	0.02
best_1000	1131	283	0	0.01	0.02
final_best	2068	518	0.02	0.03	0.03

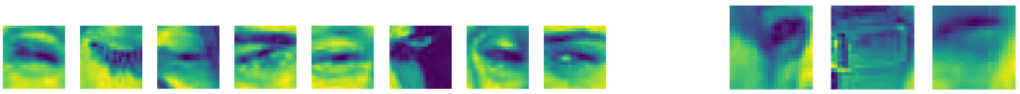
Таблица 2: Результаты каждой модели на своей выборке



(a) Глаз открыт (b) Глаз закрыт

Рис. 6: Правильные предсказания final_best на тестовой выборке

Также смотря на ошибки final_best можно заметить, что эти случаи действительно сложные и возможно все еще остались некоторые проблемы в разметке датасета. На некоторых ошибочных предсказаниях кажется, что модель на самом деле права.



(a) Глаз открыт (b) Глаз закрыт

Рис. 7: Ошибочные предсказания final_best на тестовой выборке