

Raport: 2

Projekt: KLASYFIKACJA CYFR

Wydział: Inżynierii Mechanicznej i Robotyki

Kierunek: Inżynieria Akustyczna

Rok: 3

Przedmiot: Technologia mowy

Grupa: 3

Imię i nazwisko:

1. Oliwia Chmura
2. Mateusz Juszcak
3. Henryk Kosała

1. Status pracy

1.1. Status wykonanej pracy na dzień 26.11.2025.

Zrealizowane zadania:

1. Plan pracy drugiego etapu.
2. Poprawiono czytelność kodu i zwracanych wartości w konsoli
3. Zaktualizowano funkcje:
 - get_mfcc()
 - load_mfcc_params()
 - load_load_train_files_and_determine_mfcc()
 - prepare_training_data()
 - train_gmms()
 - classifier()
4. Dodano nową funkcję do etapu ekstrakcji cech MFCC:
 - load_gmm_params(), która odpowiada za wczytanie parametrów GMM
5. Usunięto funkcję load_mfcc_data()
6. Dodano nową funkcję do etapu przygotowania danych:
 - save_processed_dataset(), która zapisuje przetworzony dataset do pliku
 - validate_data_quality(), która sprawdza dane i je zwraca w terminalu
 - save_gmm_models(), która zapisuje do pliku pickle
 - load_gmm_models(), która wczytuje gmm z pliku pickle
 - save_best_parameters(), która zapisuje parametry do pliku csv
 - load_best_parameters(), która wywołuje funkcje load_mfcc_params() i load_gmm_params()
7. Zaktualizowano funkcję test() i zmieniono jej nazwę na calculate_accuracy()
8. Dodano nową funkcję do etapu krosvalidacji - cross_validation(), która wykorzystuje KFold, a następnie trenuje modele i zwraca otrzymane wyniki
9. Dodano funkcje do etapu analizy:
 - evaluate_system, która zwraca otrzymane dane z ewaluacji
 - print_confusion_matrix(), która macierz pomyłek
10. Dodano etap optymalizacji – funkcja optimize_parameters_full().
11. Dodano funkcje etapowe, które wywołują zaimplementowane wcześniej funkcje dla danych etapów:

- ETAP: 1 – prepare_data_stage()
- ETAP: 2 – quick_prototype(), optimize_parameters_stage(), cross_validation_stage()
- ETAP: 3 – train_final_model()
- ETAP: 4 – evaluate_system_stage()

12. Przetworzono main.

13. Dogranie zbioru ewaluacyjnego

1.2. Zadania do wykonania

Zadania niezrealizowane:

1. Plan pracy trzeciego etapu projektu
2. Udoskonalenie i optymalizacja
3. algorytmu
4. Poprawa działania funkcji main()

2. Wyniki

Otrzymane wyniki przy zastosowaniu funkcji optimize_parameters_full(), w której za pomocą pętli tworzymy zbiór do pliku csv z najlepszymi parametrami. Wyniki widoczne na rysunku 2.1. Optymalna konfiguracja uzyskała dokładność 80%, przy parametrach podanych poniżej.

```

=== NAJLEPSZE PARAMETRY ===
Skuteczność: 80.00%
Parametry MFCC: {'n_mfcc': 13, 'n_fft': 1024, 'win_length': 512, 'hop_length': 200, 'n_mels': 32, 'count_delta': True, 'count_delta_delta': True}
Parametry GMM: {'n_components': 12, 'covariance_type': 'tied', 'max_iter': 200, 'random_state': 42}
Najlepsze parametry zapisano do plików:
- best_mfcc_params.csv
- best_gmm_params.csv

=== TOP 10 KOMBINACJI ===

```

| Rank | n_mfcc | n_fft | win_len | hop_len | n_mels | delta | delta2 | GMM_comp | GMM_cov | Accuracy% |
|------|--------|-------|---------|---------|--------|-------|--------|----------|---------|-----------|
| 1 | 13 | 1024 | 512 | 200 | 32 | ✓ | ✓ | 12 | tied | 80 |
| 2 | 13 | 1024 | 512 | 160 | 32 | ✓ | X | 12 | diag | 76.67 |
| 3 | 20 | 1024 | 512 | 160 | 32 | ✓ | X | 12 | diag | 76.67 |
| 4 | 13 | 512 | 400 | 160 | 32 | ✓ | ✓ | 12 | diag | 73.33 |
| 5 | 13 | 512 | 512 | 200 | 32 | ✓ | ✓ | 8 | tied | 73.33 |
| 6 | 13 | 1024 | 512 | 200 | 32 | ✓ | ✓ | 4 | diag | 73.33 |
| 7 | 16 | 512 | 400 | 160 | 32 | ✓ | X | 12 | tied | 73.33 |
| 8 | 16 | 1024 | 400 | 200 | 26 | ✓ | X | 4 | tied | 73.33 |
| 9 | 20 | 512 | 400 | 200 | 32 | ✓ | ✓ | 8 | diag | 73.33 |
| 10 | 20 | 1024 | 512 | 200 | 26 | ✓ | ✓ | 12 | diag | 73.33 |

Rysunek 2.1. Otrzymane wyniki