Rozpoznawanie mówców metodą *i*-wektorów/PLDA na urządzeniach mobilnych

Autor: Radosław Białobrzeski
Opiekun: prof. Andrzej Pacut

Seminarium Zespołu Biometrii i Uczenia Maszynowego,
24.10.2017

Cel pracy

Celem pracy jest system realizujący niezależne od tekstu, elastyczne rozpoznawanie tożsamości na urządzeniu mobilnym w oparciu o nagrania głosu. W systemie wykorzystano nowoczesne metody rozpoznawania na bazie *i*-wektorów/PLDA i zaimplementowano go w postaci przenośnej biblioteki *RVLib*. Wyniki na dwóch bazach danych porównano z metodami naukowymi oraz jedną komercyjną.

Znaczenie rozpoznawania mówców

Mobilne uwierzytelnianie z użyciem cech biometrycznych staje się codziennością:

- prawie 700 dostępnych modeli smartfonów z czytnikiem odcisku palca[1],
- Apple Pay wspierający rozpoznawanie twarzy,
- postępująca normalizacja (m.in. ISO i NIST).

Biometria głosu jako łatwa w użyciu, akceptowalna i coraz dokładniejsza to doskonałe narzędzie do mobilnego uwierzytelniania biometrycznego.

[1] https://www.91mobiles.com/list-of-phones/phones-with-fingerprint-scanner

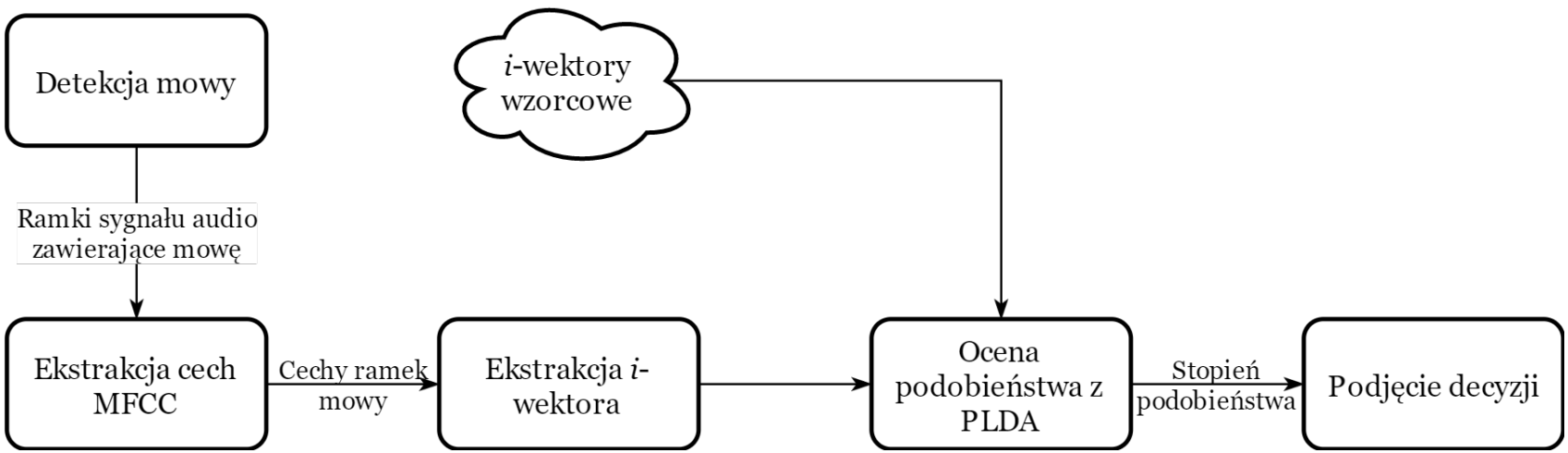
Rozwój dziedziny

Na przestrzeni lat modelowanie mówców pokonało długą drogę w kierunku maksymalnej generalizacji i niewrażliwości na zakłócenia:

1. Kwantyzacja wektorowa (Soong, 1985).
2. Modele mieszanin gaussowskich (Reynolds, 1995).
3. Adaptowane modele mieszanin gaussowskich (Reynolds, 2000).
4. Modele *i*-wektorowe (Kenny, 2008).
5. **PLDA (Kenny, 2010)**.
6. Głębokie sieci neuronowe (mowa - Hinton, 2012).

Struktura ramowa systemu

System powinien być możliwie efektywny, a przy tym nie obciążać nadmiernie procesora urządzenia mobilnego. Zdecydowano się na podejście *i*-wektorowe/PLDA.



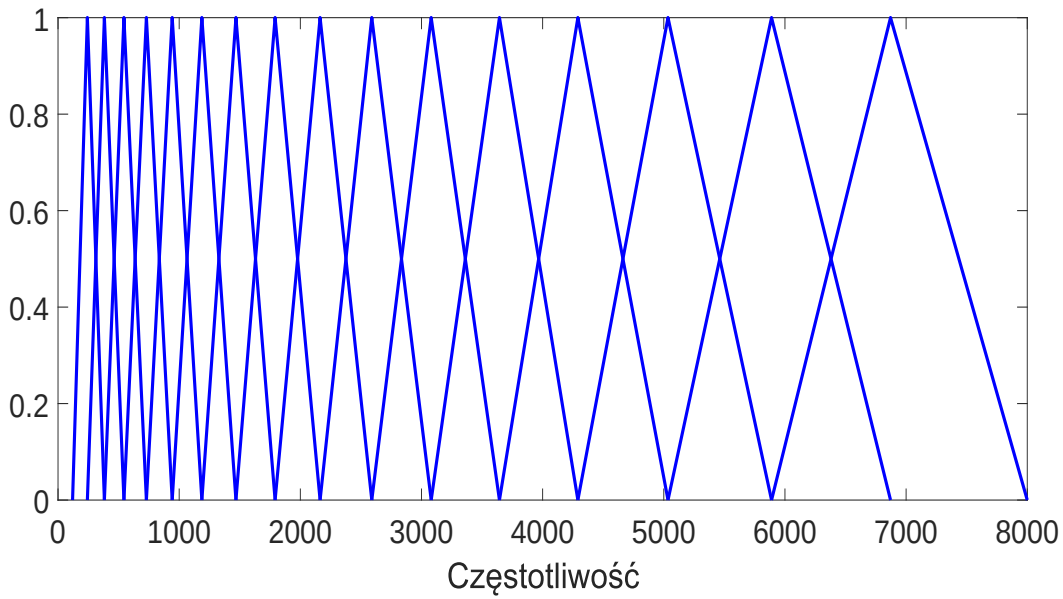
Detekcja mowy - *SSVAD*

Przy założeniu, że zakłócenia są stacjonarne (tzn. ich charakterystyka nie zmienia się w czasie), detekcję mowy zrealizować można w następujący sposób:

1. Oszacuj średnie spektrum częstotliwościowe szumu na podstawie pierwszej 0.5s nagrania.
2. W każdej ramce (aplikować okno Hamminga) sygnału:
   - 2.1 Oblicz SNR *a posteriori* dla ramki.
   - 2.2 Odejmij spektrum szumu od spektrum sygnału zaszumionego odwrotnie proporcjonalnie do SNR. Jeśli wystąpiło szczególnie niskie SNR - ustal w ramce stały, niewielki szum.
3. Zrekonstruuj sygnał przy pomocy odwrotnej transformaty Fouriera i odwrotnego okna Hamminga.
4. Ustal próg detekcji na podstawie średniej amplitudy ramek tła i szczytowych.
5. Porównaj wygładzone amplitudy ramek z progiem i określ ramki zawierające mowę.

Ekstrakcja cech MFCC

1. Podziel sygnał na ramki i przemnóż je przez okno Hamminga.
2. Wykonaj FFT na ramkach.
3. Wyznacz spektrogram - podnieś do kwadratu moduły współczynników rozwinięcia Fouriera.
4. Zaaplikuj trójkątne *Mel*-filtry.
5. Oblicz odwrotną transformatę Fouriera z logarytmów współczynników Mel-częstotliwościowych.



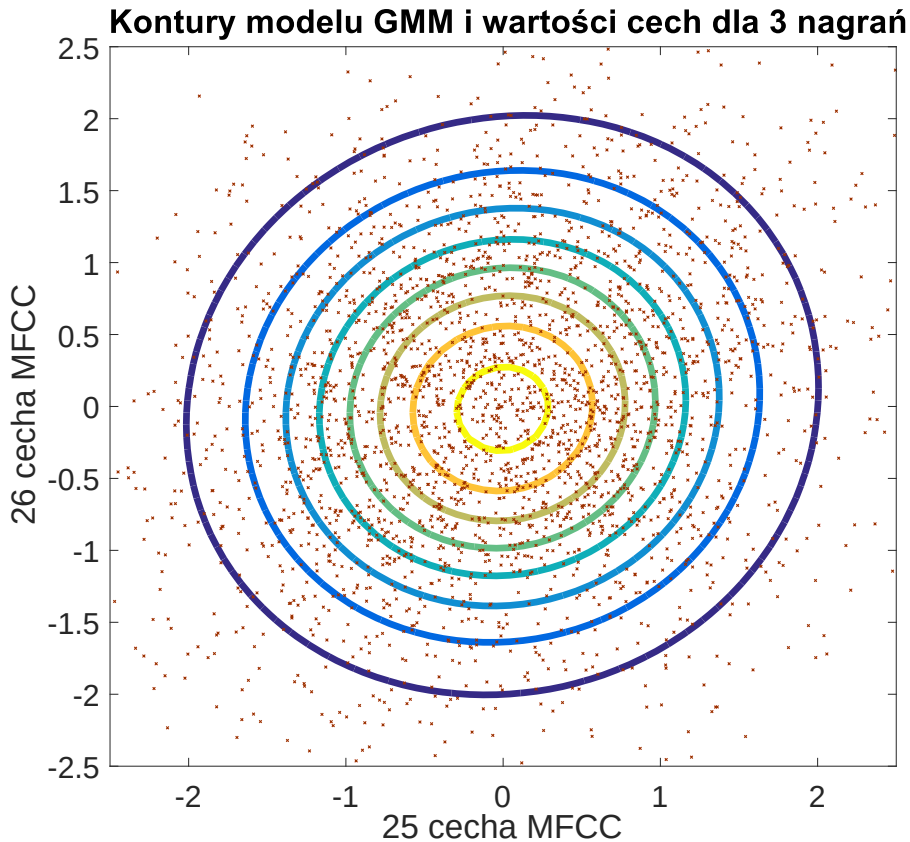
Normalizacja danych

Aby zapewnić lepsze warunki numeryczne oraz sprowadzić wektory cech MFCC bliżej rozkładów normalnych (dalej korzysta się z GMM), należy je normalizować:

- Dla każdego wektora MFCC:
  1. Oblicz średnie wartości współczynników i ich odchylenia dla okna o szerokości n.
  2. Odejmij od wektora średnie i podziel przez odchylenia standardowe.

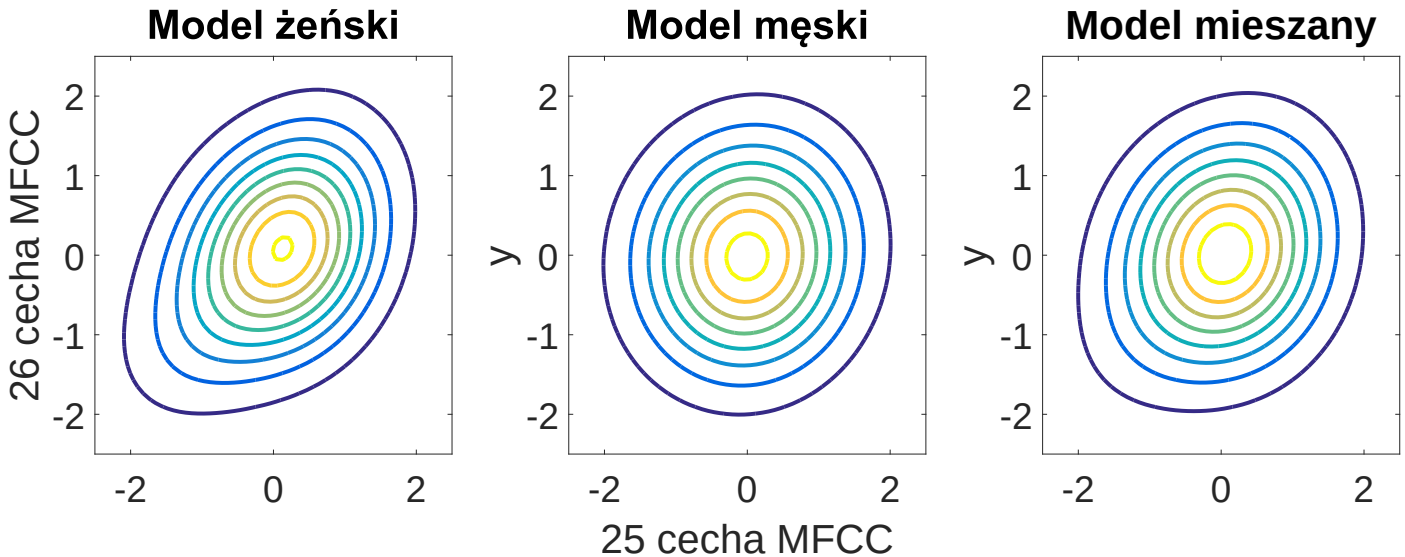
Modele tła GMM-UBM

Podstawą modelowania mówców jest tzw. **model tła**, który zwykle jest modelem mieszanin gaussowskich (GMM) nauczonym na zbiorze nagrań reprezentatywnych dla zbioru użytkowników systemu. Sposób uczenia - popularny algorytm EM, maksymalizujący w każdej iteracji warunkowe wartości oczekiwane funkcji wiarygodności.



Modele tła - problem płci

Typowo uczy się odmienne modele dla kobiet i mężczyzn.
Najczęściej powoduje to znaczną poprawę jakości modelu.



Modelowanie całkowitej zmienności

Superwektor średnich modelu GMM zależnego od mówcy wyraża się sumą superwektora średnich z modelu tła oraz macierzy całkowitej zmienności wymnożonej przez tzw. *i*-wektor. *I*-wektor to punkt w podprzestrzeni największej zmienności mówców rozpinanej przez kolumny $\mathbf{T}$.

$$\mu_s = \mu_{UBM} + \mathbf{T} \cdot \mathbf{x}$$

Macierz $\mathbf{T}$ wyznacza się, korzystając z algorytmu EM. Część jego operacji bez zmian wykorzystuje się do ekstrakcji *i*-wektorów.

Normalizacja *i*-wektorów

Już na tym etapie *i*-wektory nadają się do klasyfikacji (choćby dystansem kosinusowym), jednak należy wykonać kilka operacji, aby przystosować je do pracy z PLDA:

1. Centrowanie wokół średniej.
2. Normalizacja długości (dzielenie przez normę euklidesową).
3. „Wybielanie", czyli dekorelacja i sprowadzanie macierzy kowariancji do postaci jednostkowej.

PLDA - Zasada działania

Każdy *i*-wektor związany jest z oszacowaniem superwektora średnich modelu GMM zależnego od danego mówcy, z czym związana jest istotna niepewność pomiarowa. Ponadto, całkowita zmienność to także zmienność kanałów.
W przypadku sygnałów mowy szczególnie pomocna okazuje się probabilistyczna liniowa analiza dyskryminacyjna (PLDA):

$$\mathbf{x}_{ij} = \mu + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij} + \epsilon_{ij}$$
$$\mathbb{P}(\mathbf{x}_{ij}|\mathbf{h}_i, \mathbf{w}_{ij}, \theta) = \mathcal{N}_x[\mu + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij}, \Sigma]$$

Różne warianty

W praktyce upraszcza się modele PLDA, „wciągając" zmienność wewnątrzklasową do macierzy kowariancji - wcześniej tylko diagonalnej, teraz pełnej. Model generatywny przyjmuje wtedy postać:

$$\mathbf{x}_{ij} = \mu + \mathbf{F}\mathbf{h}_i + \epsilon_{ij}$$
$$\mathbb{P}(\mathbf{x}_{ij}|\mathbf{h}_i, \mathbf{w}_{ij}, \theta) = \mathcal{N}_x[\mu + \mathbf{F}\mathbf{h}_i, \Sigma]$$

Dokładnie zapisany algorytm EM dla tej postaci PLDA - Sizov, A. (2014) *Unifying Probabilistic Linear Discriminant Analysis Variants in Biometric Authentication*

Ocenianie podobieństwa

Prawdopodobieństwo, z jakim dwa *i*-wektory należą do jednej klasy to:

H_0 - ta sama klasa, H_1 - różne (*jakiekolwiek*) klasy

$$r_{lin}(\mathbf{x}_1, \mathbf{w}_t) = \frac{\mathbb{P}(\mathbf{w}_1, \mathbf{x}_t | H_0)}{\mathbb{P}(\mathbf{w}_1 | H_1)\mathbb{P}(\mathbf{x}_t | H_1)}$$

$$r_{log}(\mathbf{x}_1, \mathbf{x}_t) = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_t^T \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} + \mathbf{F}\mathbf{F}^T & \mathbf{F}\mathbf{F}^T \\ \mathbf{F}\mathbf{F}^T & \mathbf{\Sigma} + \mathbf{F}\mathbf{F}^T \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_t \end{bmatrix}$$

$$- \mathbf{x}_1^T \left[\mathbf{\Sigma} + \mathbf{F}\mathbf{F}^T \right]^{-1} \mathbf{x}_1 - \mathbf{x}_t^T \left[\mathbf{\Sigma} + \mathbf{F}\mathbf{F}^T \right]^{-1} \mathbf{x}_t + C$$

Wymagające założenia modelu

Założenie dot. *i*-wektorów $x \sim \mathcal{N}(...)$ nie jest spełnione nawet po normalizacji. Ponadto, do porównań N:1 wykorzystywane jest założenie o niezależności *i*-wektorów referencyjnych.

Należy więc porównywać *i*-wektory 1:1 i agregować wyniki porównań jednej osoby. Niestety, $x_1 \neq x_2 \implies \mathbb{E}r(x_1) \neq \mathbb{E}r(x_2)$ i $\mathbb{E}r^2(x_1) \neq \mathbb{E}r^2(x_2)$. Należy zatem normalizować wyniki.

Normalizacja symetryczna - modyfikacja

Schemat dwuwymiarowej kohortowej normalizacji symetrycznej *s*-norm:

$$r_{norm}(x_{i,j}) = \frac{r(x_{i,j}) - \mu_{i,*}}{\sigma_{i,*}} + \frac{r(x_{i,j}) - \mu_{*,j}}{\sigma_{*,j}}.$$

Można go zmodyfikować (s^*-norm), aby kolejne składniki sumy były obliczane sekwencyjnie, jako funkcje wyników poprzednich kroków. Otrzymuje się lepsze upodobnienie do siebie średnich i wariancji $r(x)$ dla danych x:

	bez norm.	*s*-norm	s^*-norm
$\sigma(\bar{r})$	4.39	0.22	0.0
$\sigma(s^2(r))$	60.59	0.57	0.0

Bazy danych

W pracach wykorzystywano dwie bazy danych:

- MOBIO (konkurs ICB2013) - uczenie, testowanie.
- MobiBits (**baza własna**)- testowanie.

W bazie MobiBits znajduje się prawie 5 godzin nagrań głosu pracowników NASK. Baza była pobierana latem 2017. Dane zbierano z użyciem smartfona Huawei P9 Lite w ramach czterech sesji - każda sesja miała swój indywidualny charakter.

Dobór hiperparametrów

Hiperparametry systemu dobierano przeszukiwaniem siatki parametrów (ang. *grid search*) i wyznaczaniem współczynnika EER. Uczono modele niezależne od płci z powodu niewielkiego zróżnicowania danych uczących.

Dobrano trzy zestawy hiperparametrów, maksymalizujące efektywność dla każdej z baz i zastosowania mobilnego. Bardzo dobre wyniki pokazują zarówno możliwość dopasowania systemu do bardzo szczególnego typu danych, jak i osiągnięcia satysfakcjonującej generalizacji i wydajności.

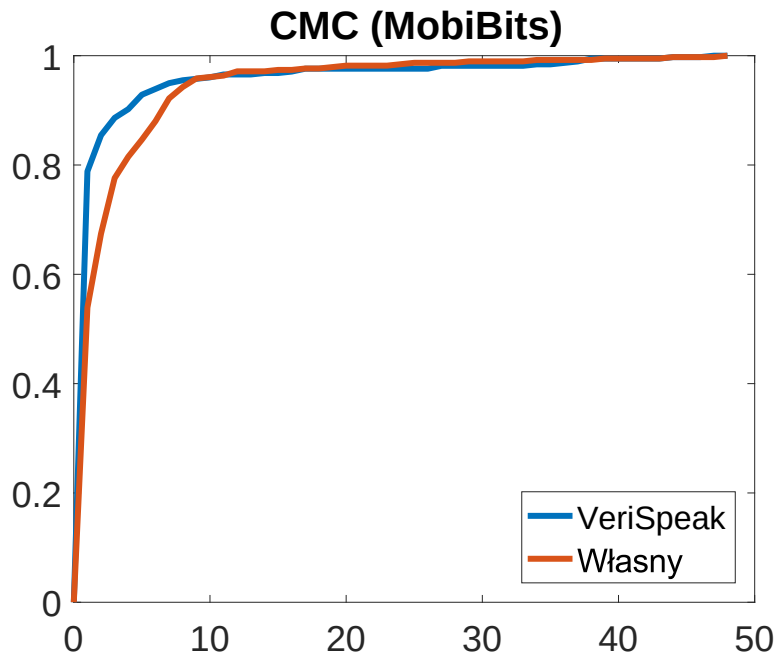
Dane konkursowe - ICB2013

Porównanie na zbiorze DEV bazy ICB2013 dla obu płci.
Hiperparametry: 128 komponentów GMM-UBM, 64 kolumny w TVM, 64 kolumny w PLDA.

	ATVS	**CDTA**	**GIAPSI**	**EHU**
EER M/F	14.88/16.83	12.73/19.47	9.68/11.59	11.31/17.93
	IDIAP	**M-T**	**Phonexia**	**RUN**
EER M/F	9.96/12.01	10.19/11.43	9.60/8.36	24.64/25.4
	Własny			
EER M/F	12.52/8.65			

Dane własne - MobiBits

Porównanie na sesjach 3 i 4 bazy MobiBits. Porównanie z Neurotechnology VeriSpeak 10.0 (339∼859€/stanowisko). Hiperparametry: 256 komponentów GMM-UBM, 64 kolumny w TVM, 64 kolumny w PLDA.



	Własny	VeriSpeak
EER	10.94%	13.60%
Rank 1	54%	78.8%
Rank 5	84.6%	92.9%

C# i Math.NET Numerics

Poza skryptami pakietu MATLAB, system rozpoznawania mowy zaimplementowano w postaci biblioteki DLL języka C#. Wykorzystano bibliotekę Math.NET Numerics - bibliotekę do obliczeń numerycznych ogólnego przeznaczenia.

```
Vector<double> scoresH1 = referenceIVectors.Transpose()
    .Multiply(PLDAQHat).Multiply(referenceIVectors).Diagonal();
double scoresH2 = testIVector.ToRowMatrix().Multiply(PLDAQHat)
    .Multiply(testIVector.ToColumnMatrix()).Diagonal().At(0);
Vector<double> scoresH1H2 = referenceIVectors.Transpose().Multiply(2.0)
    .Multiply(PLDALambda).Multiply(testIVector.ToColumnMatrix()).Column(0);
Vector<double> scores = scoresH1H2.Add(scoresH1).Add(scoresH2);
```

Niestety, biblioteka Math.NET ma liczne wady. Wydajność platformy jest niska, a kod słabo zoptymalizowany.

Biblioteka RVLib

Utworzona biblioteka jest przenośna (Windows, Linux, Android, iOS, MacOS) i wykorzystuje modele nauczone w środowisku MATLAB. Funkcje biblioteki:

- odczyt nagrań,
- detekcja mowy,
- pozyskiwanie *i*-wektorów,
- porównywanie *i*-wektorów,
- agregacja i normalizacja wyników.

Utworzono również aplikację demonstracyjną.

Aplikacja demonstracyjna

Prezentacja zewnętrzna.
Model - **zaledwie** 64 komponenty GMM-UBM, 32 kolumny w TVM i 32 kolumny w PLDA. Jakość rozpoznawania - wysoka.

Podsumowanie

Zaimplementowano działający system rozpoznawania mówców, również w wersji mobilnej. System nie ma wielkich wymagań sprzętowych ani pamięciowych, jednak cechuje się bardzo dobrą jakością rozpoznawania. Przyszłe prace:

- Detekcja mowy - niewrażliwy detektor oparty o sieci splotowe.
- Normalizacja wyników - analityczne obliczanie wartości oczekiwanej i wariancji porównań danego *i*-wektora ze wszystkimi innymi. Wyeliminowanie zapotrzebowania na kohorty.
- Optymalizacja - próby modyfikacji kodu Math.NET, aby obliczenia probabilistyczne były wykonywane szybciej.

Bibliografia

- Kasprzak, W. (2009) *Rozpoznawanie obrazów i sygnałów mowy*
- Mak, M., Yu, H. (2013) *Robust Voice Activity Detection for Interview Speech in NIST Speaker Recognition Evaluation*
- Reynolds, D.A., Rose, R.C. (1995) *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*
- Kenny, P. (2005) *Eigenvoice Modelling With Sparse Training Data*
- Shepstone, S.E. et al. (2016) *Total Variability Modelling using Source-specific Priors*
- Prince, S.J.D., Elder, J.H. (2008) *Probabilistic Linear Discriminant Analysis for Inferences About Identity*
- Sizov, A. et al. (2014) *Unifying Probabilistic Linear Discriminant Analysis Variants in Biometric Authentication*
- Rajan P. et al. (2014) *From Single to Multiple Enrollment i-vectors: Practical PLDA Scoring Variants for Speaker Verification*