

Ćwiczenie 4

KLASYFIKATOR STATYSTYCZNY

Zakres pracy

W ramach ćwiczenia należy do dostarczonego interfejsu, umożliwiającego wyświetlanie obrazów zawartych w plikach graficznych, dodać możliwość rozpoznawania wczytywanych obrazów za pomocą:

- 1) tzw. naiwnego klasyfikatora Bayesa, w którym jako cechy wykorzystuje się jasności kolejnych pikseli,
- 2) klasyfikatora statystycznego, działającego w oparciu o cechy wyznaczone za pomocą dyskretnej transformacji kosinusowej (DCT).

Należy również zaimplementować możliwość testowania wsadowego, z użyciem wszystkich obrazów znajdujących się we wskazanym folderze. Wynikiem ma być wartość błędu rozpoznawania.

Informacje pomocnicze

Naiwny (inaczej normalny) klasyfikator Bayesa jest klasyfikatorem statystycznym, tj. działającym wg następującej reguły decyzyjnej:

$$x^*(\mathbf{y}) = x_i, \quad \text{jeżeli} \quad p(x_i, \mathbf{y}) > p(x_j, \mathbf{y}) \\ j = 1, 2, \dots, L, \quad j \neq i. \quad (1)$$

Jednocześnie przyjmujemy „naiwne” założenie, że poszczególne cechy obrazu y_1, \dots, y_N są od siebie niezależne, czyli:

$$p(x_i, y_1, \dots, y_N) = p(x_i)p(y_1 | x_i)p(y_2 | x_i)p(y_3 | x_i)\dots \quad (2)$$

Często przyjmuje się, że poszczególne cechy mają w klasach rozkład normalny, dający się opisać za pomocą wartości średniej oraz wariancji. Wówczas możemy wykorzystać wzór:

$$p(y_j | x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(y_j - \mu_i)^2}{2\sigma_i^2}\right) \quad (3)$$

gdzie μ_i – średnia wartość cechy y_j w klasie x_i , σ_i – odchylenie standardowe cechy y_j w klasie x_i . Należy pamiętać, że formalnie wzór (3) opisuje funkcję gęstości prawdopodobieństwa, nie samo prawdopodobieństwo, w związku z tym może on zwracać wartości większe od 1 (co nie wpływa jednak na ostateczną decyzję). Oprócz tego, w praktyce założenie o normalnym rozkładzie wartości cech może czasami okazywać się fałszywe, stąd należy narzucić dolną granicę odchylenia standardowego: $\sigma_i > 0,1$.

Klasyfikator statystyczny działa w oparciu o następującą regułę decyzyjną:

$$x^*(\mathbf{y}) = x_i, \quad \text{jeżeli} \quad p(\mathbf{y} | x_i)P(x_i) > p(\mathbf{y} | x_j)P(x_j) \\ j = 1, 2, \dots, L, \quad j \neq i. \quad (4)$$

gdzie $P(x_i)$ oznacza prawdopodobieństwo a priori klasy x_i . Oprócz tego korzystamy z wielowymiarowego rozkładu normalnego:

$$p(\mathbf{y}|x_i) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_i|}} \exp\left\{-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{y} - \boldsymbol{\mu}_i)\right\} \quad (5)$$

gdzie $\boldsymbol{\mu}_i$ to wektor średni, a Σ_i – macierz kowariancji i -tej klasy. Macierz kowariancji obliczamy korzystając ze wzoru:

$$\Sigma_i = \frac{1}{m_i} \sum_{k=1}^{m_i} (\mathbf{y}[k] - \hat{\boldsymbol{\mu}}_i) (\mathbf{y}[k] - \hat{\boldsymbol{\mu}}_i)^T \quad (6)$$

Bezpośrednie stosowanie klasyfikatora statystycznego wiąże się z trudnościami, dlatego obrazy należy wstępnie przekształcić przy użyciu dyskretnej transformaty kosinusowej (DCT).

Wskazówki implementacyjne

Klasyfikator powinien uwzględniać 10 klas obrazów (cyfry od 0 do 9). W przypadku klasyfikatora naiwnego zbiór uczący każdej klasy powinien składać się z 5 przykładów. Dane służące do uczenia i testowania znajdują się w plikach `cyfry1.zip` (klasyfikator naiwny) i `cyfry2.zip` (klasyfikator statystyczny).

OBRAZY NALEŻY ZMNIJSZYĆ DO ROZDZIELCZOŚCI **10x10** PIKSELI.

W przypadku klasyfikatora statystycznego zmniejszamy rozdzielczość po przekształceniu DCT.

Wczytane obrazy reprezentowane są przez macierze typu `CV_8UC1`, czyli zawierające elementy typu `BYTE` o wartościach z zakresu `0..255`. Jasność obrazu powinna być jednak sprowadzona do przedziału `[0,1]`.

Zamiana macierzy **A** na macierz **B** w formacie `CV_32F` (elementy typu `float`):

```
A.convertTo(B, CV_32F);
```

Przeskalowanie macierzy *MatSrc* do rozmiarów (*szerokość* × *wysokość*) i umieszczenie wyniku w macierzy *MatDst*:

```
resize(MatSrc, MatDst, cvSize(szerokosc, wysokosc));
```

Typ `Scalar` jest dostosowany do obsługi obrazów wielokanałowych i stanowi tablicę 4 liczb typu `double`. Jeśli korzystamy z obrazów jednokanałowych (w odcieniach szarości), interesuje nas tylko element o indeksie 0.

Następująca funkcja oblicza średnią m i odchylenie standardowe s z elementów macierzy A (m i s są typu `Scalar`):

```
meanStdDev(A, m, s);
```

Do wyznaczania transformaty kosinusowej służy funkcja `dct`:

```
dct(A, A);
```

Wczytywanie obrazu z pliku wykonuje funkcja `imread`:

```
imread(sciezka_do_pliku, CV_LOAD_IMAGE_GRAYSCALE);
```