Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

Wydział Informatyki

UNL FOURIER

Karol Działowski

nr albumu: 39259
przedmiot: Ekstrakcja cech

Szczecin, 11 stycznia 2021

Spis treści

1 Cel laboratorium

Celem laboratorium było przeprowadzenie ekstrakcji cech dla logotypów samochodów korzystając z deskryptora UNL-F.

2 Wyznaczanie wektora cech

Pierwszym etapem deskryptora jest wyznaczenie macierzy przy pomocy deskryptora UNL [1].

Macierz wyjściowa tworzona jest jako zależność odległości konturu od wyznaczonego centroida względem kąta. Elementy konturu są uporządkowane w kolejności zgodnej z wska-

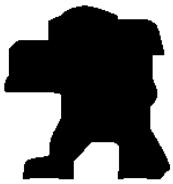
zówkami zegara. Pierwszy element konturu wyznaczany jest za pomocą przeszukiwania obrazu od góry do dołu, od lewej do prawej strony. Pierwszy piksel będący konturem zostaje elementem startowym sygnatury. Macierz wynikowa jest skalowana do rozmiaru 128×128 pikseli.

Wektor cech tworzony jest jako wycinek kwadratu o danym boku z macierzy uzyskanej z UNL. Widmo macierzy obliczane jest przy użyciu funkcji *np.fft.fft2* [2]. Wycięty kwadrat reprezentowany jest w formie wektora. Kod obliczający podany deskryptor pokazano na listingu 1.

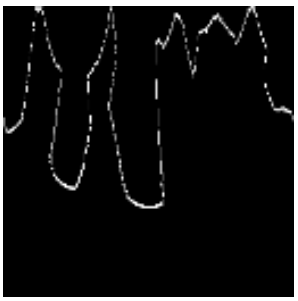
Wyznaczony deskryptor UNL przedstawiono na rysunku (1).

Kod źródłowy 1: Wyznaczanie deskryptora UNL-F
Źródło: Opracowanie własne

```
def unl(img):
    contour = object_contour(img)
    thresh = cv2.canny(img, 30, 200)
    x, y = center_of_contour(img, contour)
    (x, y), radius = cv2.minenclosingcircle(contour)
    polar_image = cv2.linearpolar(
        thresh, center=(x, y), maxradius=radius, flags=cv2.warp_polar_linear
    )
    polar_image = polar_image.astype(np.uint8)
    scaled_polar_image = cv2.resize(polar_image, (128, 128),
interpolation=cv2.inter_linear)
    result = cv2.rotate(scaled_polar_image, cv2.cv2.rotate_90_counterclockwise)
    return result


def unl_fourier(img, size, whole=false):
    if whole:
        polar_image = unl_whole(img)
    else:
        polar_image = unl(img)

    img_fft = np.fft.fft2(polar_image)
    spectrum = np.log(1 + np.abs(img_fft))

    result = list(spectrum[:size, :size].flat)
    return result
```



(a) Kształt wejściowy



(b) Macierz UNL

(c) Macierz UNL z wypełnieniem kształtu

Rysunek 1: Wyznacznia macierzy UNL

3 Klasyfikacja

Klasyfikację przeprowadza się analogicznie względem poprzednich laboratoriów, czyli tworząc słownik wzorców na podstawie obrazów uczących i w procesie predykcji wyszukuje się najbliższy wzorzec za pomocą metryki euklidesowej. Kod prezentujący proces uczenia i predykcji klasyfikatora przedstawiono na listingu 2.

Kod źródłowy 2: Proces uczenia i predykcji klasyfikatora
Źródło: Opracowanie własne

```
class UNLFClassifier(BaseEstimator, ClassifierMixin):
    def __init__(self, size=5):
        self.classes_ = None
        self.template_dict_ = None
        self.size = size

    def fit(self, X, y):
        self.classes_ = np.unique(y)

        unlf_descriptors = []
        labels = []

        for i in range(len(X)):
            im = cv2.imread(X[i], cv2.IMREAD_GRAYSCALE).astype("uint8")
            unlf_desc = unl_fourier(im, self.size)
            unlf_descriptors.append(unlf_desc)
            labels.append(y[i])

        data = {"unlf": unlf_descriptors, "label": labels}

        df = pd.DataFrame.from_dict(data)
        self.template_dict_ = df

    def predict(self, X):
        y_pred = []
        for i in range(len(X)):
            x = X[i]
```

```
            im = cv2.imread(x, cv2.IMREAD_GRAYSCALE).astype("uint8")
            descriptors = unl_fourier(im, self.size)
            y_pred.append(self.closest_template(descriptors))
        return y_pred

    def closest_template(self, descriptors):
        template_descriptors = self.template_dict_["unlf"].tolist()
        distances = cdist([descriptors], template_descriptors).mean(axis=0)
        closest_label = self.template_dict_.iloc[distances.argmin()]["label"]
        return closest_label
```

4 Wyniki

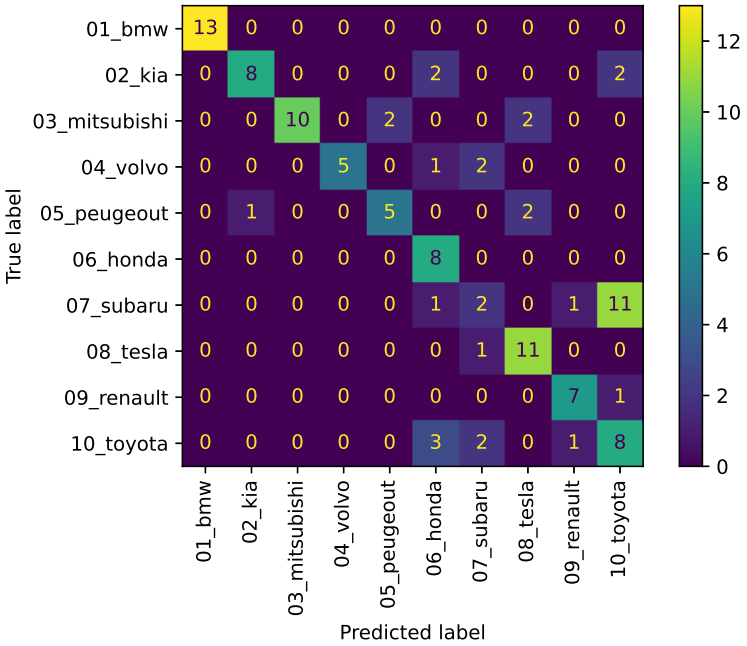
Badane rozmiary bloku: $5, 20, 30, 35, 40$. Uzyskane wyniki przedstawiono w tabeli 1.

Rozmiar bloku	Dokładność UNL-F	Dokładność UNL-F z wypełnieniem
5	0.821	0.687
20	0.758	0.723
30	0.776	0.660
35	0.714	0.767
40	0.732	0.678

Tabela 1: Eksperymenty badania rozmiaru bloku w klasyfikacji logotypów samochodu przy pomocy UNL-F

Dokładność stworzonego klasyfikatora, działającego na podstawie UNL-F, kształtuje się na poziomie 82% dla bloku o rozmiarze 5. Na obrazie (2) przedstawiono macierz konfuzji dla badanego klasyfikatora.

Modyfikacja z wypełnieniem kształtu osiąga zauważalnie gorsze wyniki od klasycznego UNL-F.



Rysunek 2: Macierz konfuzji

5 Podsumowanie

Podczas laboratorium zaimplementowano deskryptor UNL-F. Uzyskano zadowalające wyniki osiągające ponad 80% dokładności.

Charakterystyka obiektu badawczego, czyli logotypów samochodów, jest trudna dla deskryptorów kształtu. Jest to spowodowane podobieństwem kształtów pomiędzy logotypami, które przedstawiono w poprzednich sprawozdaniach.

Pewne pary logotypów są trudne do rozpoznania przez człowieka tylko na podstawie kształtu. Są to na przykład logotypu okrągłe lub o kształcie elipsoidy. Z tego powodu wyniki na poziomie 80% są zadowalające.

Bibliografia

[1] Frejlichowski D.: Contour objects recognition based on unl-fourier descriptors, pp. 203–208, sty. 2005, DOI: 10.1007/0-387-23484-5_20.

[2] Oliphant T. E.: *A guide to numpy*, Trelgol Publishing USA, 2006.