Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

Wydział Informatyki

POLAR-FOURIER GREYSCALE DESCRIPTOR

Karol Działowski

nr albumu: 39259
przedmiot: Ekstrakcja cech

Szczecin, 21 stycznia 2021

Spis treści

1 Cel laboratorium

Celem laboratorium była implementacje Polar-Fourier Greyscale Descriptor [1] w uproszczonej formie i przetestowanie go na wybranych zbiorze danych, w moim przypadku były to logotypy samochodów.

2 Implementacja deskrytptora

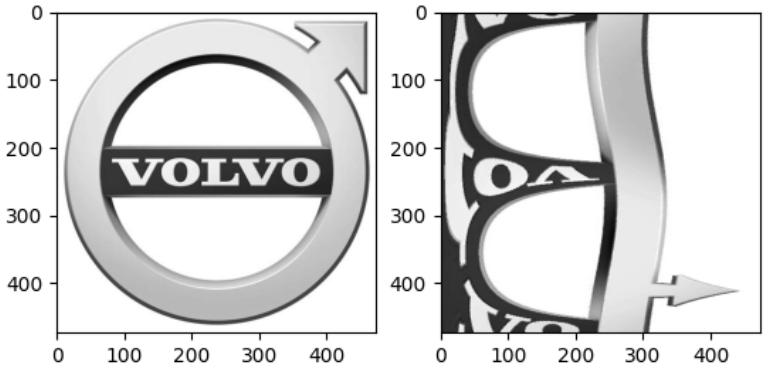
Deskryptor Polar-Fourier Greyscale Descriptor w uproszczonej formie polega na przekształceniu obrazu do skali szarości, wyznaczenia centroidu jako punkt ciężkości, transformacji do układu biegunowego i wycięcia fragmentu widma po dwuwymiarowej transformacji Fouriera [1]. W oryginalnym deskryptorze oprócz tego pojawiają się wstępne etapy, takie jak filtracja, które pominięto w tej implementacji.

Kod źródłowy 1: Wyznaczanie Polar-Fourier Greyscale Descriptor
Źródło: Opracowanie własne

```
def grayscale_fourier_desc(filename, size):
    img = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
    # Przekształcenie do skali szarości i do współrzędnych biegunowych
    max_radius = np.sqrt(((img.shape[0] / 2.0) ** 2.0) + ((img.shape[1] / 2.0) **
2.0))
    moment = cv2.moments(img)
    x = int(moment["m10"] / moment["m00"])
    y = int(moment["m01"] / moment["m00"])
    centroid = (x, y)
    polar_image = cv2.linearPolar(
        img, centroid, max_radius, cv2.WARP_FILL_OUTLIERS
    )
    polar_image = polar_image.astype(np.uint8)

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

    # Wycinek fouriera
    img_fft = np.fft.fft2(polar_image)
    spectrum = np.log(1 + np.abs(img_fft))
    result = list(spectrum[:size, :size].flat)
    return result
```



Rysunek 1: Transformacja do współrzędnych biegunowych

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 TemplateClassifier(BaseEstimator, ClassifierMixin):
    def __init__(self, descriptor, args = {}):
        self.classes_ = None
        self.template_dict_ = None
        self.descriptor_ = descriptor
        self.args_ = args

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

        features = []
        labels = []

        for i in range(len(X)):
            image_path = X[i]
            feature = self.descriptor_(image_path, **self.args_)
            features.append(feature)
            labels.append(y[i])

        data = {"feature": features, "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]
            features = self.descriptor_(x, **self.args_)
            y_pred.append(self.closest_template(features))
        return y_pred

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

4 Wyniki i wnioski

Przebadane różne wycinki widma po tranformacji 2D Fourier.

Rozmiar wyciętego bloku	Dokładność klasyfikacji
2×2	52.67%
3×3	50%
5×5	75 %
7×7	65.17 %
10×10	70.53 %

Tabela 1: Badanie rozmiaru wyciętego bloku z widma

Najlepsze wyniki osiągnięto dla rozmiaru bloku 5×5 dając dokładność na poziomie 75%.

Porównano zaimplementowany deskryptor do wcześniej omawianych deskryptorów kształtu. Wszystkie porównania przeprowadzono na tej samej zasadzie klasyfikacji, czyli porównaniu do wzorca za pomocą metryki euklidesowej. Deskryptor Polar-Fourier Greyscale Descriptor wypada gorzej tylko od UNL-F.

Deskryptor	Dokładność klasyfikacji
Polar-Fourier Greyscale Descriptor	75%
UNL-F	82.1%
Sygnatura	45%
2D Fourier	46.4%

Tabela 2: Porównanie dokładnośći klasyfikacji dla wybranych deskryptorów

Bibliografia

[1] Frejlichowski D.: Application of the polar–fourier greyscale descriptor to the automatic traffic sign recognition, *International Conference Image Analysis and Recognition*, pp. 506–513, 2015.