

ЗАСТОСУВАННЯ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ ВИЯВЛЕННЯ ДЕФЕКТІВ ВИРОБІВ ЗА ДОПОМОГОЮ АНАЛІЗУ ТЕКСТУРИ

Малішевський В. С., Ковалюк Д. О.

КПІ ім. Ігоря Сікорського, boomcrouser@gmail.com

Комп'ютерний зір, як технологія створення штучних систем, вивчає техніки та алгоритми обробки зображень і відео. Прикладами таких систем можуть бути автономні транспортні засоби, промислова робототехніка, системи відео спостереження, людино-машинні системи. В їх основу покладено задачі автоматичного розпізнавання обличчя, виявлення тих чи інших об'єктів, визначення їх ознак, класифікації зображень, тощо.

Математичною основою методів, що використовуються в задачах комп'ютерного зору є лінійна алгебра, матричні обчислення, математична статистика, методи оптимізації.

Окремими напрямом використання систем комп'ютерного зору є виявлення дефектів для різних об'єктів і процесів. Розвиток автоматизації і зменшення впливу людського фактору, поки не дозволяє повністю вирішити проблему відхилень у виробничих процесах, які можуть спровокувати дефекти. Тому розвиток систем контролю якості продукції на основі комп'ютерного зору є безумовно перспективним і актуальним.

В багатьох випадках технологія комп'ютерного зору використовується для виявлення дефектів на виробництві в режимі реального часу. Є декілька методів, які використовуються для виявлення дефектів на конвеєрі: метод аналізу текстури (використовуються для виявлення недоліків, такі як плями, подряпини, тріщини тощо), метод аналізу кольору (використовуються для виявлення дефектів, таких як відхилення від норми кольору або нерівномірність заливки), методи аналізу форми (використовуються для виявлення таких дефектів як відхилення від норми форми або нерівномірність розмірів), методи нейромереж (ці методи використовуються для автоматичного виявлення дефектів за допомогою навчання нейромережі).

В роботі розглядається задача виявлення дефекту текстури виробу на рухомому конвеєрі, що дозволяє виконувати класифікацію бракованих (дефективних) виробів на основі порівняння з валідним зразком.

Алгоритм розв'язання цієї задачі передбачає наступні етапи:

- зображення поточного та еталонного виробу необхідно перевести у відтінки сірого та застосувати для них фільтр Габо́ра;
- обчислити гістограми текстурних характеристик для обох зображень, за допомогою відстані X_i -квадрат, та нормалізувати їх;
- після нормалізації слід обчислити відстань між гістограмами текстурних характеристик.

Для проведення досліджень використано зображення, представлені на рис. 1.

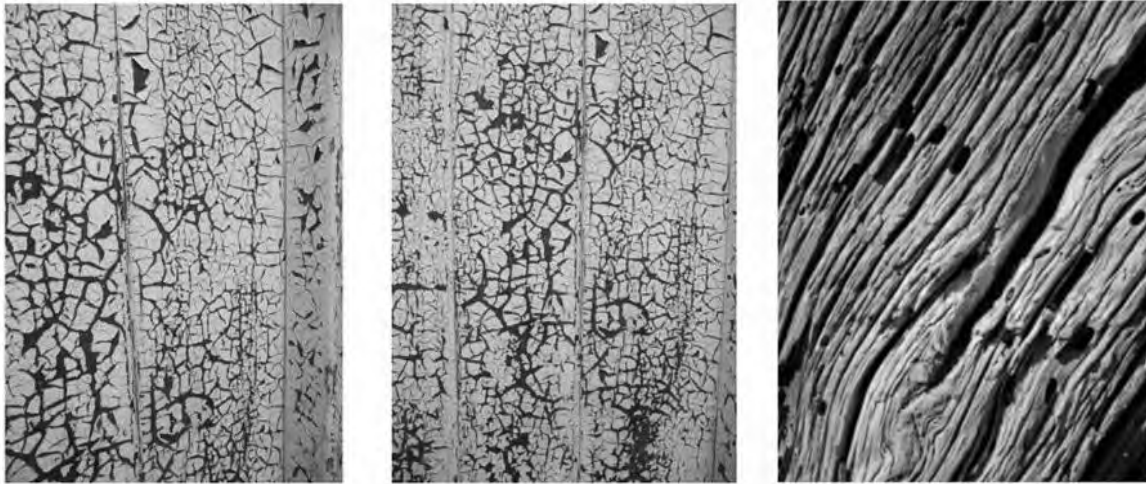


Рис 1 - Зображення текстур взятих для аналізу дефектів

Для розрахунку схожості між текстурами виконано приграмну реалізацію наведеного вище алгоритму на мові Python (фрагмент)

```
import cv2
import numpy as np
# Завантажуємо два зображення для порівняння
img1 = cv2.imread('image1.jpg')
img2 = cv2.imread('image2.jpg')
# Конвертуємо зображення в чорно-білий формат
gray_img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
gray_img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
# Застосовуємо фільтр Габора до обох зображень
gabor_filter = cv2.getGaborKernel((kernel_size, kernel_size), sigma, theta,
lambda_, gamma, psi)
filtered_img1 = cv2.filter2D(gray_img1, cv2.CV_8UC3, gabor_filter)
filtered_img2 = cv2.filter2D(gray_img2, cv2.CV_8UC3, gabor_filter)
# Обчислюємо гістограми текстурних характеристик для обох зображень
num_bins = 16
hist1 = cv2.calcHist([filtered_img1],[0],None,[num_bins], [0, 255])
hist2 = cv2.calcHist([filtered_img2],[0],None,[num_bins], [0, 255])
# Нормалізуємо гістограми
hist1 = cv2.normalize(hist1, hist1).flatten()
hist2 = cv2.normalize(hist2, hist2).flatten()
# Обчислюємо відстань між гістограмами текстурних характеристик
dist = cv2.compareHist(hist1, hist2, cv2.HISTCMP_CHISQR_ALT)
print("Відстань між зображеннями: ", dist)
```

В результаті отримаємо наступні метрики: між лівим (еталонним) і середнім зображенням – 0,0022, між лівим і правим – 2, 425. Якщо значення відстані мале, то можна стверджувати що зображення схожі між собою, тобто мають схожі текстурні характеристики. Важливо зазначити, що метод порівняння зображень за текстурними характеристиками не є універсальним. Він підходить для порівняння зображень зі схожими текстурними характеристиками, але не допоможе виявити різницю між зображеннями із зовсім різними текстурами.