

Diagnostyka Systemów

Rozpoznawanie godziny na zegarach analogowych.

Ernest Lasek 235877

Agnieszka Filosek 236092

20 Styczeń 2021

Repozytorium: github.com/erni2244/Clock_date_detector

1. Problem badawczy.

Obecnie w przemyśle przykłada się dużą uwagę do jakości wytwarzanych dóbr. Spowodowane jest to potężną rywalizacją na rynku, dlatego każdy producent stara się dostarczać produkt lepszy niż konkurencja. Z tego względu stale rozwija się dział kontroli jakości. Niejednokrotnie kontrola manualna przez wykwalifikowany personel okazuje się być zbyt czasochłonna lub nieopłacalna. W takim przypadku szuka się ratunku ze strony inżynierii. Z pomocą przychodzi wtedy automatyzacja, która usprawnia cały proces. Dziedziną zajmującą się kontrolowaniem procesu oraz jego produktów jest diagnostyka systemów.

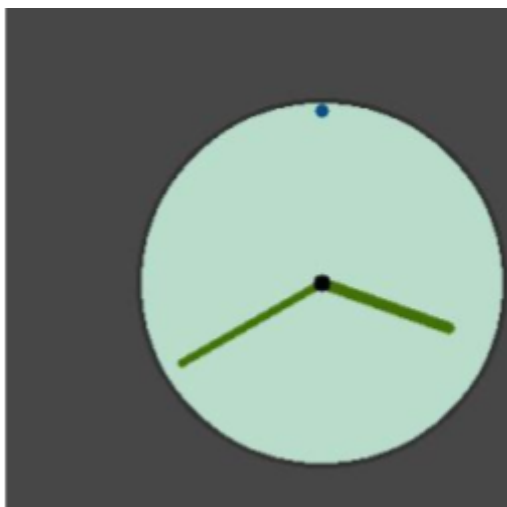
Celem omawianej pracy było rozwiązanie jednego z problemów należących do dziedziny diagnostyki systemów. Problemem tym było automatyczne weryfikowanie poprawności czasu na zegarkach analogowych. Największą trudność stanowił brak możliwości ingerowania w zegar, gdzie cały proces weryfikacji miał być zrealizowany nieinwazyjnie, jedynie obserwując obiekt za pomocą kamery.

2. Założenia projektowe.

2.1. Główne założenia.

W celu rozwiązania postawionego problemu badawczego stworzono aplikację desktopową w języku Java. Ponieważ rozpatrzenie wszystkich możliwych sytuacji mogących wystąpić było praktycznie niemożliwe, przyjęto pewne założenia projektowe. Pierwszym problemem była forma danych wejściowych. Postanowiono, że projekt zrealizowany zostanie w oparciu o dostępną w internecie gotową bazę zegarów. Z tego powodu przyjęto, że jako obraz wejściowy podawane będzie zdjęcie w formacie PNG lub JPG, o dowolnym rozmiarze, zawierające tarczę zegara analogowego na jednokolorowym tle.

Większość rozwiązań opisanych w sieci opiera się o skomplikowane metody takie jak głębokie sieci neuronowe. Często zdarza się, że jest to przerost formy nad treścią, ponieważ autorzy uciekają do nowoczesnych, skomplikowanych metod, podczas gdy stare, wielokrotnie sprawdzone metody zapewniają wystarczającą efektywność dla danego problemu. Zatem jednym z założeń projektowych było opracowanie metody, która nie bazowała by na sieciach neuronowych.



Obraz 1: Przykładowe zdjęcie

2.2. Baza danych.

Konstruując algorytm umożliwiający odczyt godziny na zegarze analogowym potrzebne były dane na których przeprowadzona została późniejsza weryfikacja poprawności działania programu. Zbiór obrazów przedstawiających zegary analogowe można było wygenerować samodzielnie, ale dane pochodzące z rzeczywistego źródła lepiej oddawały różnorodność - były ciekawsze i trudniejsze. Zatem bazę danych zastosowaną w projekcie pozyskano z kaggle, co usprawniło pracę, gdyż pobieranie obrazów zegara z Internetu jest procesem czasochłonnym. Baza ta składała się z 50 000 zegarów analogowych wraz z tłem wyposażonych w etykiety (czas na zegarze).

2.3. Narzędzia.

Do stworzenia oprogramowania spełniającego powyższe założenia, wykorzystano język programowania Java. Program napisano w środowisku IntelliJ. W celu usprawnienia pracy zespołowej korzystano z systemu kontroli wersji GitHub. Ponieważ zagadnienie obróbki zdjęć cyfrowych jest bardzo obszernym tematem, a odtwarzanie wszystkich algorytmów przetwarzania obrazów nie było celem projektu, zdecydowano się na skorzystanie z

biblioteki OpenCV. Jest to pakiet gotowych funkcji do pracy na obrazach, udostępniany na licencji „Berkeley Software Distribution Licenses”.

3. Uzyskane efekty.

3.1. Przetwarzanie wstępne danych.

Obrazy pochodzące z wykorzystanej bazy danych przedstawiały zegary analogowe. Niestety oprócz tarczy zegara posiadały one również tło, co uniemożliwiało zastosowanie ich w stanie surowym w projekcie. Zdecydowano zatem o zaimplementowaniu algorytmu przycinania obrazów. W algorytmie tym na początku ściągany był pierwszy piksel z rysunku, który uznawano jako tło. Następnie poruszano się po wierszach, a gdy nie zostało wykryte tło, tworzona była tablica tymczasowa, która zawierała informacje ile w danym wierszu tło nie wystąpiło. Tło sprawdzane było funkcją `isBackground` w której szukano piksela tła w przedziale z odchyłką 10 dla każdej barwy. Operacja ta była konieczna, ponieważ na skraju zegara pojawiały się przebarwienia. W przypadku, gdy ilość pikseli, które nie zostały rozpoznane jako tło była większa od zera zwiększano wysokość o 1. W kolejnym kroku szukano brzegu zegarka i dodawano połowę wysokości tego zegarka. W następnym etapie w analogiczny sposób wykonywano operacje dla kolumn. Aby przyciąć obrazek niezbędne było podanie od którego do którego wiersza i kolumny to skrócenie ma być przeprowadzone. Z tego względu od środka odpowiednio odejmowano i dodawano połowę z szerokości lub wysokości.

3.2. Główny algorytm.

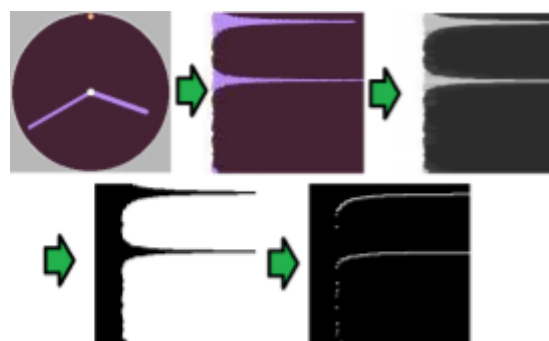
Opracowując główny algorytm odczytywania godziny na zegarze skupiono się na standardowych algorytmach, które nie wymagałyby tworzenia sieci neuronowych oraz przeprowadzania czasochłonnego

procesu ich uczenia. Po przejrzaniu różnych źródeł w Internecie oraz zapoznaniu się z dokumentacją biblioteki OpenCV, natrafiono na transformatę polarną. Przekształcała ona obiekt kołowy na zdjęciu w obiekt wertykalny. Dzięki temu możliwe było odwzorowanie tarczy zegara oraz ruchu jego wskazówek jako ruch pionowy.

Obraz poddany transformacji polarnej musiał zostać odszumiony, aby pozbyć się cyfr i pozostawić jedynie wskazówki. W tym celu najpierw poddano go dylatacji z prostokątnym jądrem o dużej szerokości. Następnie przeprowadzono dwukrotne binarne odszumianie. W efekcie otrzymano czarno-białe zdjęcie z poziomymi liniami odpowiadającymi wskazówkom. Ponieważ wskazówki zegara były dość szerokie, postanowiono poddać obraz transformacji Sobela.

Po przeprowadzeniu całego procesu przekształcania zdjęcia należało odczytać z niego czas. W tym celu odszukano najdłuższe poziome ciągi białych pikseli, a następnie pozyskano ich współrzędne pionowe. Korzystając z zależności, że tarcza zegara po transformacji polarnej miała proporcjonalnie rozłożone godziny od góry do dołu, ustalono jaką godzinę pokazują wskazówki.

Po przeprowadzeniu kilku eksperymentów uznano, że niezbędne będzie opracowanie dodatkowego algorytmu, który oceniałby kolory na zdjęciu tak, aby dobrać odpowiednie wartości odszumiania. Algorytm analizował centralną część zdjęcia w celu ocenienia jasności tarczy zegara. W sytuacji, gdy była ona ciemniejsza od wskazówek, dokonywał inwersji kolorów. Ponadto obliczana była średnia wartość kolorów tła i wskazówek, co pozwoliło określić poziom odszumiania.



Obraz 2: Przebieg transformacji obrazu

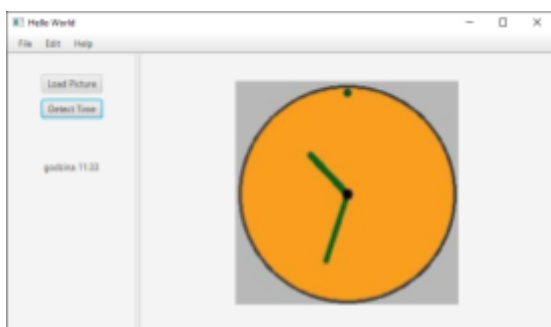
3.3. Wyniki i efektywność.

Stworzona aplikacja pozwoliła na wczytanie wybranego zdjęcia, a następnie wyświetlenie odczytanej godziny. Aplikację w ostatecznej formie przetestowano na obrazach zawartych w bazie danych. Większość wyników była poprawna z dokładnością do kilku minut. Największy problem stwarzały zegary, które miały bardzo mały kontrast pomiędzy kolorem wskazówek a kolorem tarczy. Problematyczne było również ustalenie odpowiedniego progu przejścia między poszczególnymi godzinami, dlatego zdecydowano się na rozbudowę algorytmu o filtr Sobela. Wskazówki zegara w bazie danych wskazywały dokładne godziny, pomijając przestrzeń pomiędzy, zatem algorytm musiał zaokrąglać wartości w górę. Filtr Sobela umożliwił wskazanie jednej krawędzi wskazówki, dzięki czemu uzyskano delikatne przesunięcie. Zabieg ten wyeliminował w większości przypadków omawiany błąd.

Program testowano również na kilku rzeczywistych obrazach zegarów. Potrafił on prawidłowo rozpoznać godzinę, jednak tutaj jego skuteczność było o wiele mniejsza. Poniżej można zaobserwować wyniki prawidłowo odczytanych godzin na zegarach analogowych przez program.



Obraz 3: Przykładowe działanie programu (rozpoznano 7:46)



Obraz 4: Przykładowe działanie programu (rozpoznano 11:33)

4. Wnioski.

Pomyślnie zrealizowano cel omawianej pracy, jakim było automatyczne weryfikowanie poprawności czasu na zegarkach analogowych. Mimo napotkanych trudności takich jak brak możliwości ingerowania w zegar czy konieczność ustalenia odpowiedniego progu przejścia między poszczególnymi godzinami udało się zaimplementować program, który w większości przypadków prawidłowo odczytuje godzinę na zegarze analogowym z dokładnością do 5 minut. Projekt można w przyszłości rozwijać o inne funkcjonalności oraz usprawniać algorytm weryfikowania godziny minimalizując odchyłkę błędu w odczycie.

5. Bibliografia.

- OpenCV documentation index (<https://docs.opencv.org/>)
- Baza Danych Zegarów (<https://www.kaggle.com/shivajbd/analog-clocks>)

| | |
|-------------------------------|----------|
| Problem badawczy. | 1 |
| Założenia projektowe. | 1 |
| Główne założenia. | 1 |
| Baza danych. | 2 |
| Narzędzia. | 2 |
| Uzyskane efekty. | 2 |
| Przetwarzanie wstępne danych. | 2 |
| Główny algorytm. | 2 |
| Wyniki i efektywność. | 3 |
| Wnioski. | 4 |
| Bibliografia. | 4 |