

Raport

Odtwarzanie plików zniszczonych fizycznie

2019-02-26

Autorstwa:

Dominik Albinia

Leszek Milasz

Julia Jasińska

1. Podstawowe założenia.....	2
2. Obecnie wykonana praca.....	3
3. Pozostała część pracy do wykonania.....	5
4. Podsumowanie.....	6

Podstawowe założenia

Założeniem projektu jest stworzenie programu komputerowego, który będzie w stanie odtworzyć zniszczony fizycznie dokument przez narzędzie tnące. Mogą to być narzędzia typu niszczarka do papieru, nożyczki etc.

Projekt polega na opracowaniu rozwiązań programistycznych pozwalających na odtworzenie dokumentów z danymi tekstowymi. System za pomocą sztucznej inteligencji oraz specjalnie dopasowanych algorytmów ma pozwolić na połączenie zniszczonych fragmentów tekstu ze sobą i odtworzenie pierwotnej części z najmniejszą możliwą utratą informacji.

Na początku algorytm będzie zajmował się tekstem w języku angielskim, a później, jeżeli będzie możliwość, w języku polskim.

Systemem operacyjnym docelowo ma być Windows oraz Linux. Program w podstawowych założeniach ma być napisany w języku Python oraz Java.

Podstawowym narzędziem dla języka Python oraz Java będzie OpenCV. Podstawowym celem projektu będzie formatowanie zniszczonych fragmentów tekstu oraz odpowiednie ich przetwarzanie do nierelacyjnej bazy danych. System ma zebrać wszystkie możliwe informacje na temat tekstu, który został wczytany.

Ważnym przyjętym przez nas założeniem jest zebranie informacji o tym, czy dany fragment posiada również fragmenty tekstu i jeżeli ma, to sprawdzenie z której strony na danym fragmencie znajduje się część tekstu.

Zostało to wzięte pod uwagę, ponieważ pierwszym etapem łączenia tekstu będzie sklejanie ze sobą odpowiednio przeciętych fragmentów liter bądź cyfr.

Dla przykładu:

- jeżeli na jednym fragmencie tekstu po lewej stronie przeczytamy fragment litery 'a' oraz uda nam się znaleźć kolejny fragment, na którym po prawej stronie znajdziemy resztę fragmentu litery 'a', to z dużym prawdopodobieństwem będziemy mogli założyć, że te dwa fragmenty będzie można połączyć. W ten sposób nie będzie potrzebne sprawdzenie tych fragmentów pod względem gramatycznym, co może zmniejszyć czas oczekiwania na zakończenie algorytmu.

W taki sposób moglibyśmy połączyć szczególne fragmenty z czasem $O(n)$ co skróciłoby czas na wykonanie reszty algorytmu.

Obecnie wykonana praca

Obecnie wykonano:

- Podzielono się swoją wiedzą na temat uczenia maszynowego.
- Dokonano podsumowania posiadanej wiedzy oraz podzielono się obowiązkami na odpowiednie etapy czytania tekstu.
- Dokonano analizy w jaki sposób czytać informacje o każdym fragmencie. Przykładowo dane mogą być przekazywane do bazy danych w postaci pliku JSON zawierający informacje o:
 - Zawartym tekście
 - Informacje o 8 krawędziach na temat fragmentu litery bądź cyfry, szczególne zacięcia, itp. (lewa górna krawędź, lewa po środku, lewa dolna krawędź, górna środkowa, prawa górna krawędź, prawa po środku, prawa dolna krawędź, dolna środkowa)
 - Wielkość fragmentu
 - Informacje czy jest już połączony na jednej z 4 części
 - Informacje na temat kąta nachylenia tekstu względem osi OX i OY
 - Inne dodatkowe informacje
- Dokonano przemyśleń na temat algorytmu sprawdzającego gramatykę
- Określono plan na następne tygodnie
- Ze strony Dominika Albiniaka:
 - Wykonano algorytm czytania tekstu oryginalnego oraz odtworzonego po zniszczeniu oraz porównanie zgodności pomiędzy nimi. Wynik z wykonanego kodu jest liczbą zmiennoprzecinkową z zakresu [0, 1]. Wynik będzie zwracany komputerowi tak, by 'uczyć' program coraz lepszego łączenia fragmentów. Wykonano w języku Java.
- Ze strony Leszka Milasza:
 - Rozpoczęcie pracy nad algorytmem rozpoznającym fragmenty liter, przy założeniu wcześniejszego wypoziomowania ścinków oraz znajomości użytej czcionki.

- Przegląd algorytmów użytecznych jako potencjalne klasyfikatory oraz sposobów ich użycia w celu uzyskania linii tekstu.
- Ze strony Julii Jasińskiej:
 - Porównano istniejące rozwiązania rozdzielające obraz na części w zależności od wielkości (równe/nierówne części) oraz rodzaju przecięcia (rzędy/kolumny/prostokąty oraz proste linie/nieregularne kształty)
 - Rozpoczęcie prac nad napisaniem własnego algorytmu w języku Python który dzieli obraz na daną liczbę podobnej wielkości części używając nieregularnych kształtów jako linii przecięcia.

Pozostała część pracy do wykonania

Pozostała część pracy do wykonania:

- Stworzenie 'dema' algorytmu łączącego tekst za pomocą gramatyki
- Stworzenie algorytmu do odpowiedniego fragmentowania zdjęcia wejściowego
- Stworzenie sieci neuronowej
- Połączenie wspólnego kodu i przetestowanie kompatybilności
- Stworzenie nierelacyjnej bazy danych (zalecane mongoDB)
- Obsługa czytania tekstu, którego fragment został przechylony względem osi OX i OY
- Obsługa polskich znaków
- Obliczenie złożoności pesymistycznej, wartości oczekiwanej oraz wariancji

Podsumowanie

Ze względu na duży nakład pracy, który wciąż pozostaje, przewidywany czas, w którym zaprezentujemy pierwsze demo to koniec marca lub początek kwietnia. Dużym wyzwaniem w projekcie wciąż pozostaje nauczanie maszynowe oraz użycie każdej dostępnej informacji w celu jak najszybszego przywrócenia pierwotnego tekstu zniszczonego fizycznie.

Wciąż pozostaje kwestia użycia OpenCV. Wraz z odpowiednim postępowaniem w tworzeniu własnych algorytmów, część rozwiązań OpenCV może zostać zastąpiona własnymi rozwiązaniami.

W przeciągu najbliższych tygodni, prosilibyśmy o dalsze omówienie koncepcji naszych rozwiązań oraz zaleceń dotyczących poprawy algorytmów.