**Przetwarzanie Języka Naturalnego**

**Sprawozdanie z projektu**

**K. Z.**

**K. W.**

**D. Z.**

**Cel Projektu**

Celem projektu jest stworzenie narzędzia, które zautomatyzuje proces identyfikacji i klasyfikacji hieroglifów egipskich z obrazów.

**Zakres Projektu**

* **Generowanie zbioru danych i jego przetwarzanie**
  + Generowanie zbioru danych składającego się z hieroglifów z listy Gardinera.
  + Przetwarzania obrazu, polegające na wycięciu indywidualnego hieroglifu z obrazu.
* **Rozwój Modelu Uczenia Maszynowego**
  + Opracowanie i trenowanie modelu sieci neuronowej do klasyfikowania hieroglifów na podstawie przetworzonych obrazów oraz testowanie i dostosowywanie modelu do poprawy jego dokładności i niezawodności.
* **Stworzenie Interfejsu Użytkownika**
  + Projektowanie i implementacja graficznego interfejsu użytkownika (GUI) w Tkinter, umożliwiającego łatwe ładowanie, wyświetlanie i tłumaczenie obrazów hieroglifów.

**Metodyka**

* Zdefiniowanie wymagań systemu oraz określenie zakresu projektu na samym początku pracy.
* Przygotowanie danych potrzebnych do realizacji projektu
  + Generowanie zbioru danych
  + Odpowiednie przetworzenie wygenerowanych obrazów
* Wykorzystanie metod uczenia maszynowego, umożliwiających poprawną identyfikację poszczególnych hieroglifów przez model sieci neuronowej.
* Przeprowadzenie licznych testów w celu optymalizacji modelu.
* Podsumowanie wyników i sporządzenie dokumentacji

**Użyte Narzędzia**

**Język Programowania - Python**: ze względu na jego popularność w przetwarzaniu danych i uczeniu maszynowym, Python jest idealnym wyborem dla tego rodzaju projektu.

**TensorFlow:** otwartoźródłowa biblioteka do uczenia maszynowego opracowana przez Google Brain Team. Jest jedną z najbardziej popularnych i zaawansowanych bibliotek w dziedzinie uczenia maszynowego i głębokiego uczenia.

**Keras:** wysokopoziomowy interfejs API (Application Programming Interface) dla sieci neuronowych, który został zintegrowany z TensorFlow, jedną z najbardziej zaawansowanych bibliotek do uczenia maszynowego. Jego celem jest umożliwienie szybkiego eksperymentowania z sieciami głębokimi.

**OpenCV:** potężna biblioteka do przetwarzania obrazów i wizji komputerowej. Jest ona szeroko stosowana w aplikacjach rzeczywistości rozszerzonej, robotyce, systemach bezpieczeństwa i w wielu innych obszarach.

**PIL:** jedna z najbardziej popularnych bibliotek w Pythonie służącą do otwierania, manipulowania i zapisywania wielu różnych formatów plików obrazowych. W naszym projekcie PIL odgrywa kluczową rolę w przetwarzaniu i analizie obrazów.

**NumPy:** biblioteka wykorzystywana do obliczeń naukowych w Pythonie. Jest niezwykle ważna w wielu dziedzinach analizy danych, naukowych obliczeń, inżynierii i uczenia maszynowego, zwłaszcza w projektach związanych z przetwarzaniem obrazów.

**Tkinter:** standardowa biblioteka GUI (graficznego interfejsu użytkownika) dla Pythona. Jest oparta na Tk, popularnym narzędziu do tworzenia interfejsów użytkownika w wielu językach programowania.

**Opis działania naszego programu**

Nasz system rozpoczyna działanie od aktywacji graficznego interfejsu użytkownika, zbudowanego przy użyciu biblioteki Tkinter, która stanowi pierwszy punkt kontaktu z użytkownikiem.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Następny etap obejmuje ładowanie kluczowych komponentów: zaprojektowanego modelu sieci neuronowej oraz leksykonu, używanego w dalszym procesie tłumaczenia.

W module gui.py zaimplementowano funkcje pozwalające na wybór i wyświetlanie obrazu w interfejsie użytkownika.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Po wybraniu obrazu przez użytkownika, system, korzystając z funkcji zaimplementowanej w module cropping.py, rozpoczyna przetwarzanie obrazu.

Obraz zawierający zrzut ekranu, Czcionka, tekst

Opis wygenerowany automatycznie

Dzięki algorytmom przetwarzania obrazu z OpenCV, system identyfikuje i wycina poszczególne hieroglify poprzez wykrywanie ich kontur. Każdy hieroglif jest następnie przekształcany i normalizowany do odpowiedniego formatu, kompatybilnego z modelem sieci neuronowej.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Model sieci neuronowej, który został wytrenowany i zaimplementowany przy użyciu Keras i TensorFlow, klasyfikuje każdy hieroglif, przypisując mu etykietę zgodnie z nauczoną klasyfikacją.

**Model Sieci Neuronowej**

**Generowanie datasetu**

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

**Tworzenie i trenowanie modelu**

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Obraz zawierający tekst, dokument, numer, Czcionka

Opis wygenerowany automatycznie

**Testowanie modelu**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, diagram, zrzut ekranu, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie



**Zapis modelu**



**Architektura modelu**

Obraz zawierający tekst, zrzut ekranu, menu, Czcionka

Opis wygenerowany automatycznie

W kolejnym kroku, wykorzystujemy wcześniej załadowany leksykon tłumaczeń, aby przekonwertować zidentyfikowane etykiety hieroglifów na odpowiadające im teksty.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Ostatni etap działania naszego systemu polega na prezentacji wyników tłumaczenia w graficznym interfejsie użytkownika. Wykorzystuje on funkcje z gui.py do wyświetlenia zarówno oryginalnych hieroglifów, jak i ich przetłumaczonego odpowiednika, zapewniając użytkownikowi kompleksowy wgląd w znaczenie analizowanego obrazu.

Obraz zawierający tekst, zrzut ekranu, diagram, Czcionka

Opis wygenerowany automatycznie

**Podsumowanie**

Naszym celem było stworzenie aplikacji, która jest w stanie odczytywać hieroglify z obrazów, klasyfikować je i tłumaczyć na tekst. Za pomocą zaawansowanego modelu sieci neuronowej oraz metod przetwarzania obrazu, udało nam się zrealizować ten cel. Aplikacja potrafi skutecznie wykrywać hieroglify oraz dokładnie je klasyfikować, a następnie tłumaczyć na odpowiadający im tekst przy użyciu wcześniej załadowanego leksykonu tłumaczeń.

**Bibliografia**

* **Lista hieroglifów: https://en.wikipedia.org/wiki/List\_of\_Egyptian\_hieroglyphs**
* **Dokumentacja OpenCv: https://docs.opencv.org/4.x/**
* **Dokumentacja Tensorflow: https://www.tensorflow.org/api\_docs**
* **Dokumentacja Tkinter: https://docs.python.org/3/library/tkinter.html**