

KNN-Math: Rozpoznávání matematických vzorců

Dokumentace projektu

13. května 2025

Obsah

1	Úvod	2
2	Struktura projektu	2
3	Popis modulů	2
3.1	Modul main.py	2
3.2	Modul preprocessing	2
3.3	Modul recognition	2
3.4	Modul parsing	2
3.5	Modul utils	3
3.6	Modul config.py	3
4	Algoritmus rozpoznávání	3
5	KNN klasifikátor	3
6	Dataset a trénování	4
7	Použití aplikace	4
7.1	Trénování modelu	4
7.2	Testování modelu	4
7.3	Rozpoznávání matematického výrazu v obrázku	4
8	Možná rozšíření	5
9	Závěr	5

1 Úvod

Tento dokument popisuje strukturu a implementaci projektu KNN-Math, který se zaměřuje na rozpoznávání matematických vzorců pomocí algoritmu k-nejbližších sousedů (KNN). Systém je navržen pro rozpoznávání jak tištěných, tak ručně psaných matematických výrazů a jejich převod do formátu LaTeX.

2 Struktura projektu

Projekt je organizován do modulárního systému, který usnadňuje údržbu a rozšiřitelnost. Hlavní adresářová struktura je následující:

3 Popis modulů

3.1 Modul `main.py`

Tento modul slouží jako hlavní vstupní bod aplikace a poskytuje rozhraní příkazové řádky pro různé operace:

- Trénování modelu pomocí trénovacích dat
- Testování modelu na testovacích datech
- Rozpoznávání matematických výrazů v jednotlivých obrázcích

3.2 Modul `preprocessing`

Tento modul se stará o přípravu vstupních obrázků pro rozpoznávání:

- **`segmentation.py`** - Obsahuje funkce pro segmentaci matematických symbolů z celého obrázku, převod obrázků do stupňů šedi, prahování pro oddělení symbolů od pozadí a normalizaci velikosti obrázků na standardní rozměr.

3.3 Modul `recognition`

Tento modul implementuje algoritmus rozpoznávání symbolů:

- **`knn_classifier.py`** - Implementuje KNN klasifikátor pro rozpoznávání jednotlivých symbolů. Zahrnuje funkce pro trénování modelu, predikci symbolů, ukládání a načítání natrénovaného modelu.

3.4 Modul `parsing`

Tento modul převádí rozpoznané symboly do LaTeX formátu:

- **`latex_converter.py`** - Obsahuje mapování mezi rozpoznávanými symboly a jejich LaTeX reprezentací, a funkci pro sestavení výsledného LaTeX kódu z posloupnosti symbolů.

3.5 Modul utils

Tento modul poskytuje pomocné funkce:

- **data_loader.py** - Funkce pro načítání obrázků a popisků z datasetu.
- **evaluation.py** - Funkce pro vyhodnocení přesnosti a výkonnosti modelu.

3.6 Modul config.py

Konfigurační soubor obsahující globální parametry:

- Cesty k adresářům (dataset, modely)
- Parametry KNN klasifikátoru (počet sousedů, váhy)
- Parametry pro předzpracování obrázků (velikost po normalizaci)

4 Algoritmus rozpoznávání

Algoritmus rozpoznávání matematických výrazů v aplikaci probíhá v následujících krocích:

1. **Načtení vstupního obrázku** - Obrázek je načten a převeden do stupňů šedi.
2. **Segmentace symbolů** - Obrázek je prahován pro oddělení symbolů od pozadí. Následně jsou nalezeny kontury jednotlivých symbolů a extrahovány jako samostatné obrázky.
3. **Předzpracování symbolů** - Každý extrahovaný symbol je normalizován na standardní velikost a provedeny další úpravy pro zvýšení přesnosti rozpoznávání.
4. **Klasifikace symbolů** - KNN klasifikátor predikuje identitu každého symbolu na základě natrénovaného modelu.
5. **Konverze do LaTeXu** - Posloupnost rozpoznávaných symbolů je převedena do LaTeX formátu, který zachovává matematickou strukturu výrazu.

5 KNN klasifikátor

Pro rozpoznávání jednotlivých symbolů je použit algoritmus k-nejbližších sousedů (KNN). Princip tohoto algoritmu je následující:

- Každý obrázek je reprezentován jako vektor příznaků (v našem případě jako vektor hodnot pixelů).
- Pro klasifikaci nového symbolu je vypočtena vzdálenost mezi jeho vektorem příznaků a vektory všech trénovacích vzorků.
- Je vybráno k trénovacích vzorků s nejmenší vzdáleností.

- Nový symbol je klasifikován do třídy, která se nejčastěji vyskytuje mezi těmito k nejbližšími sousedy.

V implementaci používáme weighted KNN, kde vzdálenější sousedé mají menší vliv na konečnou klasifikaci.

6 Dataset a trénování

Pro trénování a testování modelu je použit dataset matematických symbolů. Dataset obsahuje tisíce obrázků v adresáři `dataset` a soubor `train_labels.txt`, který obsahuje mapování mezi názvy souborů a odpovídajícími symboly.

Proces trénování zahrnuje:

1. Načtení obrázků a jejich štítků
2. Předzpracování obrázků (normalizace velikosti, převod do stupňů šedi, atd.)
3. Extrakce příznaků (v našem případě příznaky jsou přímo hodnoty pixelů)
4. Trénování KNN klasifikátoru
5. Uložení natrénovaného modelu pro pozdější použití

7 Použití aplikace

Aplikaci lze použít několika způsoby:

7.1 Trénování modelu

```
python -m app.main --train
```

7.2 Testování modelu

```
python -m app.main --test
```

7.3 Rozpoznávání matematického výrazu v obrázku

```
python -m app.main --recognize cesta/k/obrazku.png --output  
vystup.tex
```

8 Možná rozšíření

Navržená struktura projektu umožňuje snadné rozšíření o další funkce:

- **Lepší algoritmy segmentace** - Implementace pokročilejších metod pro segmentaci složitějších matematických struktur (zlomky, mocniny, integrály).
- **Pokročilejší klasifikátory** - Nahrazení KNN klasifikátoru konvolučními neuronovými sítěmi nebo jinými algoritmy strojového učení.
- **Rozpoznávání struktury** - Implementace algoritmů pro rozpoznávání struktury matematických výrazů, nejen jednotlivých symbolů.
- **Webové rozhraní** - Vytvoření webového rozhraní pro snadné použití aplikace.

9 Závěr

Tato dokumentace popisuje strukturu a implementaci projektu KNN-Math. Projekt je navržen modulárně, což umožňuje snadnou údržbu a rozšiřování. Implementovaný systém používá algoritmus k-nejbližších sousedů pro rozpoznávání matematických symbolů a převod do LaTeX formátu.