A picture containing text, font, logo, graphics

Description automatically generated

Signature detection

Proiect la Procesarea Imaginilor

Bunea-Domșa Cristian Andrei

Molnár Zsanett-Ingrid

Mureșan Miruna Mioara-Ioana

Link de git https://github.com/Miruna08-01/Signature\_detection\_opencv

1. Algoritmul KNN

La baza funcționării aplicației noastre de detectare a semnăturilor stă algoritmul KNN (K-Nearest Neighbors). KNN este un algoritm de învățare automată care clasifică sau estimează valori, bazat pe vecinii cei mai apropiați dintr-un set de date.

Algoritmul KNN are următorii parametri:

* **const vector<pair<vector<Point2f>, string>>& train\_set:**

Train\_set este un vector care are înăuntrul lui un Hashmap. Hashmapul conține perechi de cheie-valoare. În cazul nostru, cheia reprezintă un vector de 12 centre de greutate (**<vector<Point2f>)** pentru semnătură și valoarea este un string, care precizează dacă semnătura este adevărată sau falsă. Acest parametru are rolul de a “învăța” programul cum să clasifice semnăturile.

* **const vector<Point2f>& test\_signature:**

Test\_signature reprezintă semnătura care urmeză să fie testată. Semnătura e reprezentată sub forma unui vector de puncte.

* **int k:**

K reprezintă numărul de vecini cei mai apropiați pe care îi ia în considerare algoritmul în procesul de clasificare. Alegerea valorii optime pentru K este importantă în funcție de specificul problemei și setul de date. Un K mic poate duce la o sensibilitate mare, în timp ce un K mare poate duce la o învățare prea generalizată și la pierderea detaliilor subtile.

1. Experimente

Scopul proiectului era să implementăm un program care să detecteze dacă o semnătură a unei persoane e adevărată sau falsificată. Pentru realizarea acestui lucru, un obiectiv secundar a fost să maximizăm acuratețea programului, găsind cea mai bună combinație de parametri și valori pe care să lucreze acesta.

La efectuarea experimentelor, principalele aspecte pe care le-am luat în considerare sunt: valoarea lui k și formula pentru calcularea distanței dintre puncte. La variarea acestor doi parametri am ajuns la niște rezultate surpinzătoare, pe baza cărora am tras concluziile cum am putea optimiza programul. Pentru experimente am ales să analizăm datele a 6 useri: User 4, User 5, User 7, User 10, User 44, User 68.

* 1. Varierea lui K

Așa cum am menționat mai sus, alegerea valorii optime pentru K e importantă. Noi am ales ca acesta să fie un întreg din intervalul [10, 15] și să fie un număr impar. E important ca acesta să fie și impar, deoarece în cazul unui număr par, programul ar putea da rezultatul de 50% adevărat și 50% fals.

* 1. Distanța Euclidiană vs distanța cosinus

La calcularea distanțelor, am folosit două formule diferite: distanța Euclidiană și distanța cosinus.

**Distanța cosinus**: pentru a calcula distanța cosinus dintre două puncte, se utilizează formula cosinusului între vectorii corespunzători acestor puncte. Se calculează produsul scalar al celor doi vectori și se împarte la produsul normelor acestora. Astfel, distanta cosinus între doi vectori este definită ca 1 minus similitudinea cosinus dintre aceștia.

**Distanța euclidiană**: pentru a calcula distanța euclidiană dintre două puncte, se măsoară lungimea liniei drepte care le leagă într-un spațiu euclidian. Distanța euclidiană este calculată ca rădăcina pătrată a sumei pătratelor diferențelor dintre coordonatele respective.

Următoarele grafice reprezintă rezulatele experimentelor noastre. Ele ilustrează atât varierea lui K, cât și calcularea distanței dintre puncte în cele 2 moduri diferite.

A screenshot of a computer

Description automatically generated with medium confidence

Rezultatele de la User 4

A screenshot of a computer

Description automatically generated with medium confidence

Rezultatele de la User 5

A screenshot of a computer

Description automatically generated with medium confidence

Rezultatele de la User 7

A screenshot of a graph

Description automatically generated with medium confidence

Rezultatele de la User 10

A screenshot of a computer

Description automatically generated with medium confidence

Rezultatele de la User 44

A picture containing text, screenshot, plot, line

Description automatically generated

Rezultatele de la User 68

* 1. Analiza rezultatelor

**User 4:** Am testat acuratețea programului pentru 3 valori ale lui K: K=11, K=13, K=15. Cu distanța Euclidiană, acuratețea pentru K=11 e 88%, pentru K=13 e 92%, iar pentru K=15 e tot 92%. Cu distanța cosinus, acuratețea pentru K=11 e 88%, pentru K=13 e tot 88%, iar pentru K=15 e 96%. Așadar, cea mai mare acuratețe o obținem pentru K=15 cu distanța cosinus, 96%.

**User 5:** La User 5 se poate observa un caz mai special. Indiferent cum variem K-ul și ce formulă folosim pentru distanță, acuratețea programului rămâne constantă, având o valoare de 80%.

**User 7:** Am testat acuratețea programului pentru 3 valori ale lui K: K=11, K=13, K=15. Cu distanța Euclidiană, acuratețea pentru K=11 e 76%, pentru K=13 e 72%, iar pentru K=15 e tot 76%. Cu distanța cosinus, acuratețea pentru K=11 e 80%, pentru K=13 e tot 80%, iar pentru K=15 e 72%. Așadar, cea mai mare acuratețe o obținem pentru K=11 și K=13 cu distanța cosinus, 80%.

**User 10:** Am testat acuratețea programului pentru 3 valori ale lui K: K=11, K=13, K=15. Cu distanța Euclidiană, acuratețea pentru K=11 e în jur de 56%, pentru K=13 e 45%, iar pentru K=15 e 60%. Cu distanța cosinus, acuratețea pentru K=11 e in jur de 52%, pentru K=13 e in jur de 36%, iar pentru K=15 e in jur de 47,5%. Așadar, cea mai mare acuratețe o obținem pentru K=15 cu distanța euclidiană, 60%.

**User 44:** Am testat acuratețea programului pentru 3 valori ale lui K: K=11, K=13, K=15. Cu distanța Euclidiană, acuratețea pentru K=11 e 80%, pentru K=13 e 76%, iar pentru K=15 e 80%. Cu distanța cosinus, acuratețea pentru K=11 e 76%, pentru K=13 e 76%, iar pentru K=15 e in jur de 84%. Așadar, cea mai mare acuratețe o obținem pentru K=15 cu distanța cosinus, 84%.

**User 68:** Am testat acuratețea programului pentru 3 valori ale lui K: K=11, K=13, K=15. Cu distanța Euclidiană, acuratețea e o valoare constantă pentru toate cele 3 valori ale lui K, și anume 84%. Cu distanța cosinus, acuratețea pentru K=11 e 76%, pentru K=13 e 80%, iar pentru K=15 e 88%. Așadar, cea mai mare acuratețe o obținem pentru K=15 cu distanța cosinus, 88%.

1. Concluzii

Obținând aceste rezultate, am constatat următoarele observații:

* Algoritmul KNN a funcționat cel mai bine pentru User 4, acesta având o acuratețe de 96%.
* Algoritmul KNN a funcționat cel mai puțin bine pentru User 10, acesta având doar o acuratețe de 60%.
* În majoritatea cazurilor, cu distanța cosinus obținem rezultate mai bune.
* Pentru userii prezenți, cea mai bună combinație de valoare - formulă este: K=15 – distanța cosinus.