# Raport tehnic

Andrei Mihai-Cosmin, Cazanov Veaceslav, Girnet Andrei, Mihalcenco David, Szocs Mihaela-Felicia

- Introducere domeniu

  Proiectul se bazează pe ideea de binarizare globală și locală, având ca scop combinarea eficientă a rezultatelor obținute de algoritmi individuali (cunoscuți și sub denumirea de praguri de binarizare). Binarizarea globală implică aplicarea unui singur prag de binarizare asupra întregii imagini, transformând pixelii în valori de alb (255) sau negru (0) în funcție de acest prag.

  Soluția primește un set de fișiere în format CSV ca intrare, fiecare conținând informații de binarizare asociate unei imagini specifice. Proiectul procesează acest set de date, combinând multiplele praguri de binarizare asociate fiecărei imagini sub forma unui arbore. Arborele are pragurile de binarizare ca frunze, iar nodurile reprezintă diverse operații programatice (if, else) pentru a realiza o combinație inteligentă a acestora. Rădăcina arborelui reprezintă pragul de binarizare final obținut în urma aplicării acestor operații.

  Dataset-ul constă în două seturi de fișiere CSV: un set "training" pentru construirea arborilor și un set "test" pentru evaluarea celor mai buni arbori creați. Fiecare set are două fișiere asociate imaginilor și ground truth-ului. Fiecare rând din fișiere conține rezultatele obținute de la diferiți algoritmi pentru o anumită imagine. Fișierele includ informații precum praguri de binarizare generate de algoritmi euristici (ex. Otsu, Niblack) și valorile F-measure asociate ground truth-ului, reprezentând performanța în intervalul 0-1. ( avem ca exemplu fig.1. si fig.2. din https://ocw.cs.pub.ro/courses/mps/proiect

- Descriere soluție

  Un algoritm potențial pentru construcția de arbori, inspirat din Monte Carlo, presupune următorii pași: se formează un arbore cu un număr aleatoriu de niveluri, situat, între x și y nivele. Pentru fiecare nivel, se stabilește un număr aleator de noduri, similar cu limita setată pentru numărul de nivele. La fiecare nivel, începând de la frunze și terminând la rădăcină, se selectează un nod cu o operație aleatoare și un număr potențial de noduri copil în funcție de operația asociată. Nodurile frunză sunt mereu alese cu o valoare asociată unui prag de binarizare specific unui anumit algoritm. Fiecare algoritm poate fi ales doar o singură dată ca frunză în întregul arbore.

- Arhitectură

  Soluția propusă pentru construirea arborilor de binarizare combină o abordare inspirată din Monte Carlo cu elemente specifice prelucrării de imagini. Arhitectura constă într-un set de pași bine definiți: Această arhitectură combină elemente de aleatorizare pentru a explora spațiul soluțiilor în mod eficient și pentru a evita convergența către o soluție locală. Rezultatele preliminare arată că această abordare poate obține scoruri de fMeasure competitive pe setul de date de antrenament.

1) Generarea aleatoare a arborelui
2) Alegerea aleatoare a operațiilor
3) Alegerea aleatoare a frunzelor
4) Procesarea imaginii

- Rezultate intermediare ale soluției software

  In general am observat ca pe fisierele de training obtinem rezultate de fMeasure in jurul cifrei 60 +- 5, care din punctul nostru de vedre e un rezultat intermediar bun. In fisierele de pe github din resources sunt 2 tree-uri care au fost cu un scor mai bun decat restul.

- Concluzii intermediare (incluzând abordarea globală și locală) In general am observat ca daca ii permitem algoritmului sa proceseze o singura imagine 0.2 sec, sa incerce diferiti arbori, cu o probabilitate de 95 la suta vom obtine cel mai bun rezulat. La implementarea pe tot setul de training am observat ca obtinem in jurul de 60 fmeasure, despre care momentan nu stim daca este o valoare buna sau nu. In general o abordare pur random nu e cea mai de succes abordare dar e una care ar putea da rezultat intr-o singura iterare sau in un infinit de iterari, asa cum am mentionat.

- Rezultate finale ale soluției software- Cu privire la rezultatul final, este de mentionat ca nu am ajuns la un fMeasure de 80, asa cum am discutat ca ar fi trebuit, in Milestone-ul precedent, ci rezultatul nostru este undeva la 74, cu aproximatie. Totodata, important de mentionat este ca am observat faptul ca, local, urmand indicatiile punctului 3 din sugestia de rezolvare ("se aplică valoarea de prag pe respectivul pixel, dacă valoarea acestuia e mai mare decât valoarea de prag obținută acesta devine alb sau ia valoarea 1, altfel devine negru sau ia valoarea 0"), uitandu-ne in resurse, exista, in resursele locale, valori luate ciudat, spre exemplu pentru un 0.95 s-a pus un 0, iar pentru un 0,4, s-a pus 1, neintelegand exact care este principiul dupa care s-a facut aceasta alegere.

- Concluzii finale- In concluzie, nu am ajuns la cele mai bune rezultate prin aceasta metoda random folosita, insa acel fMeasure de aproximativ 74, este o performanta, credem noi, in regula, tinand cont de metoda aleasa.