

Specificațiile Proiectului: White Paper Alignment based on Line and Corner Detection

1. Introducere

Acest proiect are ca scop dezvoltarea unei aplicații capabile să detecteze și să alinieze corect o foaie de hârtie albă dintr-o imagine, folosind tehnici de procesare a imaginii. Implementarea va fi realizată în **C++**, utilizând biblioteca **OpenCV** pentru detectarea marginilor, identificarea colțurilor și aplicarea transformărilor de perspectivă necesare.

2. Specificația formatului datelor

2.1 Date de intrare

- **Tip date:** O imagine digitală conținând o foaie albă, posibil cu text sau alte elemente grafice.
- **Format suportat:** `.jpg`, `.png`, `.bmp`.
- **Dimensiune:** Variabilă, însă se recomandă imagini de înaltă rezoluție pentru o detecție mai precisă.
- **Posibile distorsiuni:** Înclinare, perspectivă neuniformă, iluminare neuniformă, zgomot de fundal.

2.2 Date de ieșire

- **Imagine redresată:** Unde foaia de hârtie este aliniată corespunzător ca un dreptunghi perfect.
- **Format de ieșire:** `.jpg`, `.png`, `.bmp`.
- **Opțional:** Imagine intermediară cu colțurile detectate evidențiate.

3. Prezentarea temei

3.1 Formulare exactă a problemei

Scopul proiectului este dezvoltarea unei aplicații care să primească drept input o imagine conținând o foaie albă și să returneze o imagine unde foaia este corect aliniată. Algoritmul va detecta marginile și colțurile foi, va calcula transformarea necesară și va aplica o transformare de perspectivă pentru a obține rezultatul dorit.

3.2 Obiective

- Detectarea marginilor foi folosind metode de procesare a imaginilor.
- Identificarea colțurilor foi și ordonarea lor corespunzătoare.
- Aplicarea unei transformări de perspectivă pentru corectarea alinierii.
- Afișarea și salvarea imaginii redresate.
- Implementare eficientă în **C++** utilizând **OpenCV**.

3.3 Figuri explicative

(Se vor include diagrame/imagini ce ilustrează procesul de detecție a colțurilor și transformarea de perspectivă.)

4. Cerințe funcționale

4.1 Cerințe generale

- Aplicația trebuie să accepte imagini în formate standard.
- Detectarea colțurilor trebuie să funcționeze chiar și în condiții de iluminare variabilă.
- Rezultatul final trebuie să fie o imagine corect aliniată.
- Implementarea trebuie să fie optimizată pentru o execuție rapidă.

4.2 Funcționalități principale

1. **Încărcarea imaginii:** Utilizatorul încarcă o imagine din sistem.
2. **Preprocesare:** Convertirea imaginii în grayscale și reducerea zgomotului.
3. **Binarizare adaptivă:** Aplicarea **Adaptive Thresholding** pentru a îmbunătăți contrastul și a facilita detectarea marginilor.
4. **Detectare muchii:** Utilizarea algoritmilor OpenCV pentru identificarea marginilor foi.
5. **Detectare colțuri:** Aproximarea conturului foi și selecția colțurilor relevante.
6. **Transformare perspectivă:** Aplicarea unei transformări pentru corectarea alinierii foi.
7. **Export rezultat:** Afișarea și salvarea imaginii procesate într-un format standard.

4.3 Constrângeri

- Timpul de procesare trebuie să fie optimizat pentru a permite execuția în timp real.
- Algoritmul trebuie să funcționeze pentru foi de dimensiuni diferite și poziții variate.
- Sistemul trebuie să fie robust la variații de iluminare și zgomot de fundal.

5. Implementare în C++

5.1 Tehnologii utilizate

- **Limbaj de programare:** C++
- **Bibliotecă principală:** OpenCV pentru procesarea imaginilor
- **Mediu de dezvoltare recomandat:** Visual Studio, CLion sau CodeBlocks

5.2 Etapele implementării

1. **Încărcarea și preprocesarea imaginii**, incluzând conversia în grayscale și reducerea zgomotului.
2. **Aplicarea binarizării adaptive** pentru a obține un contrast optim între hârtie și fundal.

3. **Detectarea marginilor folosind operatorul Canny** pentru evidențierea conturilor.
4. **Identificarea conturilor și selecția celui mai mare contur dreptunghiular.**
5. **Aproximarea colțurilor folosind metode de detectare a punctelor de interes.**
6. **Aplicarea transformării de perspectivă** pentru a obține imaginea finală corect aliniată.
7. **Exportul imaginii procesate și afișarea rezultatelor.**

6. Concluzie

Acest document prezintă cerințele și specificațiile pentru dezvoltarea unei aplicații capabile să detecteze și să alinieze o foaie de hârtie dintr-o imagine, folosind **C++** și **OpenCV**.