

Proiect: Aplicație de Procesare Imagine - Mirror și Grayscale

1. Descrierea aplicației cerute

Aplicația are ca scop procesarea unei imagini preluate de la un API extern, aplicând două efecte principale:

- **Mirror** (reflexie orizontală a imaginii)
- **Grayscale** (convertirea imaginii în alb-negru, procesată pe 4 felii egale).

Aplicația este implementată folosind tehnologii web de bază: HTML, CSS și JavaScript. Ea afișează timpul de execuție al fiecărui efect în interfața utilizatorului.

2. Partea teoretică

2.1 Despre procesarea imaginilor

Procesarea imaginilor este o tehnică utilizată pentru a analiza, modifica și îmbunătăți imaginile digitale. Efectele folosite în această aplicație sunt:

- **Mirror:** Se obține prin inversarea poziției pixelilor pe orizontală. Acest proces presupune schimbarea pixelilor de pe partea stângă cu cei de pe partea dreaptă.
- **Grayscale:** Conversia pixelilor în nuanțe de gri presupune calcularea unei medii a canalelor de culoare RGB, rezultând un ton de gri uniform.

2.2 Funcționarea unui API

Un API (Application Programming Interface) permite interacțiunea dintre aplicații. În acest proiect, utilizăm API-ul [Dog CEO](#) pentru a prelua imagini aleatorii ale câinilor.

2.3 Măsurarea timpului de execuție

Pentru a evalua performanța fiecărui efect aplicat, folosim metoda `performance.now()` din JavaScript, care returnează un timp precis în milisecunde.

3. Descrierea implementării

3.1 Tehnologii utilizate

- **HTML:** Structura paginii web.
- **CSS:** Stilizarea interfeței.
- **JavaScript:** Logica aplicației, inclusiv procesarea imaginilor și afișarea datelor.

3.2 Arhitectura aplicației

- **Preluarea imaginii:** Imaginea este preluată de la API-ul Dog CEO.
- **Efecte aplicate:**
 - **Mirror:** Reflexie pe orizontală.
 - **Grayscale:** Conversia la alb-negru în 4 felii procesate secvențial.
- **Timp de execuție:** Fiecare etapă afișează timpul în interfață.

4. Descrierea funcțională a aplicației implementate

1. Interfața principală:

- a. Un buton care permite utilizatorului să încarce o imagine aleatoare.
- b. Un canvas care afișează imaginea procesată.
- c. Un spațiu dedicat pentru afișarea JSON-ului asociat imaginii și a timpilor de execuție.

2. Flux de lucru:

- a. Apăsarea butonului declanșează descărcarea imaginii de la API.
- b. Imaginea este afișată în canvas.
- c. Se aplică efectele Mirror și Grayscale, cu timpii de execuție afișați în timp real.

5. Descrierea modulelor

5.1 Preluarea imaginii

Codul JavaScript preia datele de la API și afișează informațiile în interfață.

5.2 Aplicarea efectului Mirror

Reflexia imaginii pe orizontală este realizată prin schimbarea poziției pixelilor.

5.3 Procesarea Grayscale

Imaginea este împărțită în 4 felii, fiecare fiind procesată separat pentru a converti pixelii în nuanțe de gri.

5.4 Afișarea timpilor

Timpul de execuție pentru fiecare etapă este calculat și afișat sub canvas.

6. Bibliografie

1. [MDN Web Docs - CanvasRenderingContext2D](#)
2. [JavaScript Performance.now\(\)](#)
3. [Converting Color Images to Grayscale | by 2paisay | Medium](#)
4. [javascript - HTML Canvas: How to draw a flipped/mirrored image? - Stack Overflow](#)
5. [Horizontal flip an image using canvas](#)