

# Documentatie Tema 1

## 1. Prezentare Generala

Acet proiect implementeaza un sistem de Computer Vision pentru punctarea automata a jocului de *Qwirkle Connect*. Sistemul analizeaza o serie de imagini secentiale, detecteaza configuratia tablei, identifica piesele noi plasate si calculeaza scorul conform regulamentului oficial.

## 2. Arhitectura Pipeline-ului

Procesarea fiecarei imagini urmeaza urmatorul flux secential:

1. **Preprocesare & Extragere Careu:** Identificarea tablei de joc in imaginea bruta si aplicarea corectiei de perspectiva (*Warp Perspective*).
2. **Detectie Piese:** Segmentarea pieselor de pe tabla folosind tehnici de morfologie si analiza de culoare.
3. **Clasificare:** Identificarea formei (Template Matching + Analiza Geometrica) si a culorii (Analiza HSV) pentru fiecare piesa.
4. **Mapare Matriceala:** Transpunerea pieselor detectate intr-o matrice digitala 16x16.
5. **Logica de Joc:** Compararea starii curente cu starea anterioara pentru a identifica mutarea si calculul scorului.

## 3. Module Principale si Algoritmi

### 3.1. Extragerea Tablei (extrage\_careu)

Foloseste o **abordare hibrida cu validare in 3 pasi** pentru robustete:

- **Pasul 1: Detectie Geometrica (Canny Edge):** Incearca sa gaseasca conturul tablei folosind muchiile.

- **Pasul 2: Validare (este\_rezultat\_valid):** Verifica daca conturul gasit este corect folosind:
  - *Arie:* Trebuie sa ocupe >15% din imagine.
  - *Aspect Ratio:* Trebuie sa fie aproximativ patrat (0.8 - 1.2).
  - *Culoare:* Media HUE trebuie sa fie in spectrul Verde (35-95).
  - *Contrast (StdDev):* Deviatia standard a pixelilor trebuie sa fie mare (>35) pentru a distinge tabla texturata de o masa uniforma.
- **Pasul 3: Fallback Culoare:** Daca geometria esueaza, se activeaza detectia bazata strict pe masca de culoare verde.
- **Rafinare (rafineaza\_colturi):** Odata gasit un careu brut, se aplica un padding si o masca stricta de culoare pentru a elimina marginile mesei care ar fi putut fi incluse eronat.

### **3.2. Recunoasterea Pieselor (extrag\_forme\_din\_imagine)**

Deoarece piesele sunt negre pe fundal verde, se foloseste o masca inversa:

1. Se detecteaza tot ce nu este verde/alb (adica piesele negre).
2. In interiorul fiecarei piese negre, se cauta simbolul colorat.

#### **A. Clasificare Simboluri**

Pentru a identifica cu precizie simbolurile de pe piesele de joc (Cerc, Trifoi, Romb, Patrat, Stea 4, Stea 8), sistemul utilizeaza o abordare in doua etape, combinand robustetea analizei pixel-cu-pixel cu flexibilitatea analizei geometrice.

##### **1. Preprocesare (preprocesare\_simbol)**

Inainte de orice clasificare, simbolul detectat (ROI) trece printr-un proces de normalizare:

- **Decupare:** Se elimina spatiul negru excesiv din jurul simbolului alb. Simbolul este redimensionat astfel incat cea mai mare dimensiune a sa sa incapa intr-un canvas standard (64x64 pixeli), pastrandu-si raportul original latime/inaltime.
- **Centrare si Padding:** Simbolul scalat este plasat in centrul canvas-ului negru, cu o margine de siguranta pentru a facilita suprapunerea.

## **2. Etapa I: Template Matching (*identifica\_forma\_template*)**

Aceasta este metoda principală, fiind cea mai precisa pentru formele complexe (ex: diferența dintre Trifoi și Stea 4).

- **Biblioteca de Sabloane:** Sistemul încarcă un set de imagini de referință (templates) pre-procesate identic.
- **Comparare:** Imaginea curentă a simbolului este comparată cu fiecare sablon folosind coeficientul de corelație normalizat (TM\_CCOEFF\_NORMED).
- **Invarianta la Rotatie:** Deoarece piesele pot fi rotite pe tabla, algoritmul roteste sablonul de referință la 0, 90, 180 și 270 de grade și retine cel mai bun scor de potrivire.
- **Toleranță prin Blur:** Se aplică un filtru Gaussian (GaussianBlur) atât pe sablon cât și pe imaginea tinta. Acest lucru face algoritmul mai tolerant la mici imperfecțiuni de printare sau pixeli de zgromadire.
- **Prag de Decizie (Threshold):** Dacă cel mai bun scor depășește 0.8 (80% similaritate), forma este considerată identificată pozitiv.

## **3. Etapa II: Fallback Geometric (*classify\_shape*)**

Dacă Template Matching nu găsește o potrivire clară (scor < 0.8) sau returnează "Unknown", sistemul apelează la o analiză bazată pe proprietățile conturului.

- **Analiza Conturului:** Se calculează mărimi precum Circularitatea, Convexitatea, Soliditatea și numărul de Defecți de Convexitate.
- **Decizie Logică:** Pe baza acestor mărimi, se aplică un arbore de decizie (ex: "Dacă circularitatea > 0.82, este Cerc", "Dacă are > 3 defecți de convexitate mari, este Stea").

Această metodă servește ca plasă de siguranță pentru formele deformate sau slab iluminate care nu se potrivesc perfect cu sabloanele.

## **B. Clasificare Culoare (*classify\_color*)**

- **Metoda:** Analiză HSV pe media pixelilor din simbol.
- **Calibrare:** Intervalele sunt calibrate specific pentru condițiile de iluminare.

- **Filtrare Zgomot:** Se ignora pixelii cu Saturatie mica (<50) pentru a evita confuzia dintre reflexii (alb/gri) si culori reale.

### 3.3. Logica de Joc (`calculeaza_scor_complex`)

Calculeaza punctajul conform regulilor Qwirkle:

- **Linii:** Identifica liniile orizontale si verticale formate de piesele noi.
- **Qwirkle:** Acorda +6 puncte daca o linie atinge lungimea de 6.
- **Bonusuri Tabla:** Foloseste `config_matrix` (generata la inceputul jocului) pentru a acorda puncte suplimentare pentru casutele marcate cu '1' si '2'.
- **Stabilitate:** Daca o piesa a fost detectata la mutarea T, ea este copiata automat la T+1 daca detectia esueaza momentan.

## 4. Structura Datelor

### Matricea de Stare (`stare_curenta`)

O matrice NumPy 16x16.

- None: Celula goala.
- Tuple (`ID_Forma`, `Cod_Culoare`): Celula ocupata (ex: (4, 'R')).

### Matricea de Configuratie (`config_matrix`)

O matrice NumPy 16x16 generata la startul jocului (`j_00.jpg`).

- 1: Casuta bonus un punct.
- 2: Casuta bonus doua puncte.
- -1: Diagonalele negre.

Pentru a genera matricea de configuratie, s-a utilizat Template Matching pentru detectarea pozitiilor cifrelor de "2" de pe tabla. Iar apoi, s-a construit configuratia tablei in functie de aceste pozitii.