

# Documentatie Tema 1

## 1. Prezentare Generala

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

## 2. Arhitectura Pipeline-ului

Procesarea fiecărei imagini urmează următorul flux secvențial:

1. **Preprocesare & Extragere Careu:** Identificarea tablei de joc în imaginea brută și aplicarea corecției de perspectivă (*Warp Perspective*).
2. **Detectie Piese:** Segmentarea pieselor de pe tabla folosind tehnici de morfologie și analiza de culoare.
3. **Clasificare:** Identificarea formei (Template Matching + Analiza Geometrică) și a culorii (Analiza HSV) pentru fiecare piesă.
4. **Mapare Matriceala:** Transpunerea pieselor detectate într-o matrice digitală 16x16.
5. **Logica de Joc:** Compararea stării curente cu starea anterioară pentru a identifica mutarea și calculul scorului.

## 3. Module Principale și Algoritmi

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

Folosește o **abordare hibridă cu validare în 3 pași** pentru robustețe:

- **Pasul 1: Detectie Geometrică (Canny Edge):** Încearcă să găsească 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 (extrage\_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 principala, fiind cea mai precisa pentru formele complexe (ex: diferenta dintre Trifoi si Stea 4).

- **Biblioteca de Sabloane:** Sistemul incarca un set de imagini de referinta (templates) pre-procesate identic.
- **Comparare:** Imaginea curenta a simbolului este comparata cu fiecare sablon folosind coeficientul de corelatie normalizat (TM\_CCOEFF\_NORMED).
- **Invarianta la Rotatie:** Deoarece piesele pot fi rotite pe tabla, algoritmul roteste sablonul de referinta la 0, 90, 180 si 270 de grade si retine cel mai bun scor de potrivire.
- **Toleranta prin Blur:** Se aplica un filtru Gaussian (GaussianBlur) atat pe sablon cat si pe imaginea tinta. Acest lucru face algoritmul mai tolerant la mici imperfectiuni de printare sau pixeli de zgomot.
- **Prag de Decizie (Threshold):** Daca cel mai bun scor depaseste 0.8 (80% similitudine), forma este considerata identificata pozitiv.

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

Daca Template Matching nu gaseste o potrivire clara (scor < 0.8) sau returneaza "Unknown", sistemul apeleaza la o analiza bazata pe proprietatile conturului.

- **Analiza Conturului:** Se calculeaza metrice precum Circularitatea, Convexitatea, Soliditatea si numarul de Defecte de Convexitate.
- **Decizie Logica:** Pe baza acestor metrice, se aplica un arbore de decizie (ex: "Daca circularitatea > 0.82, este Cerc", "Daca are > 3 defecte de convexitate mari, este Stea").

Aceasta metoda serveste ca plasa de siguranta pentru formele deformate sau slab iluminate care nu se potrivesc perfect cu sabloanele.

## B. Clasificare Culoare (classify\_color)

- **Metoda:** Analiza HSV pe media pixelilor din simbol.
- **Calibrare:** Intervalele sunt calibrate specific pentru conditiile 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 copiată 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.