The automation of the SET card game

Card Recognition Using Computer Vision

Wouter Kool

December 10, 2013

Contents

- Introduction: the Card Game SET
- Research question
- Method
  - Locating cards
  - Classification of the cards
  - Finding of SETs
- Results
- Implementation
- Discussion

Introduction: The Card Game SET

A SET card

- **Quantity** - 1, 2 or 3
- **Fill** - open, half-filled or solid
- **Color** - red, purple or green
- **Shape**
  - **Ravensburger** - rectangle, tilda, oval
  - **NNN Games** - diamond, tilda, rounded rectangle

Introduction: The Card Game SET

A SET card

- **Quantity** - 1, 2 or 3
- **Fill** - open, half-filled or solid
- **Color** - red, purple or green
- **Shape**
  - **Ravensburger** - rectangle, tilda, oval
  - **NNN Games** - diamond, tilda, rounded rectangle

A SET

Combination of three cards, such that each individual property equal or all different for the three cards

Introduction: The Card Game SET

A SET card

- **Quantity** - 1, 2 or 3
- **Fill** - open, half-filled or solid
- **Color** - red, purple or green
- **Shape**
  - **Ravensburger** - rectangle, tilda, oval
  - **NNN Games** - diamond, tilda, rounded rectangle

A SET

Combination of three cards, such that each individual property equal or all different for the three cards

The goal

Finding SETs amongst 12 (or more) cards

Introduction: Example of a SET

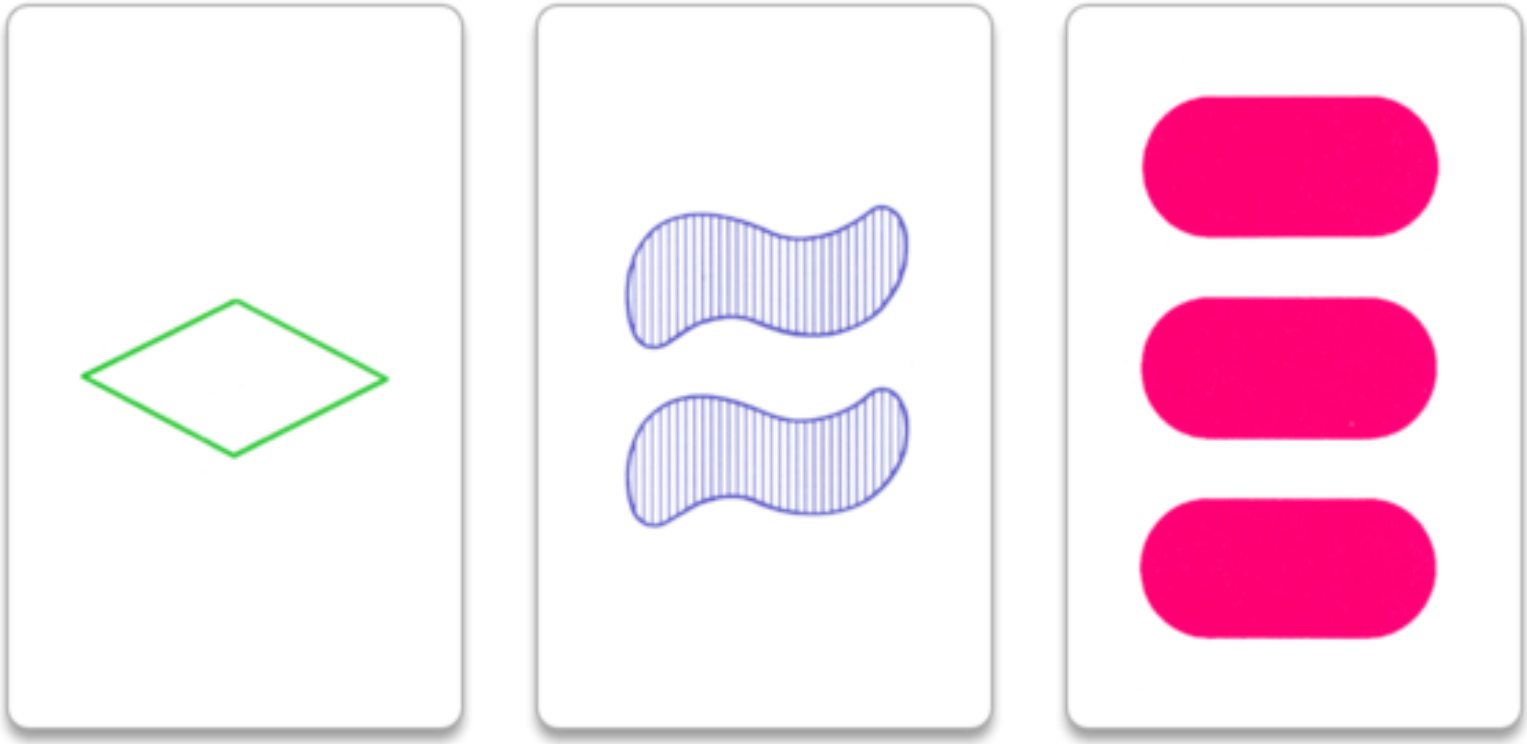


Figure : Example of a SET from the NNN Games edition.

Research question

- How can SETs automatically be detected from an image using Computer Vision?

Research question

- How can SETs automatically be detected from an image using Computer Vision?

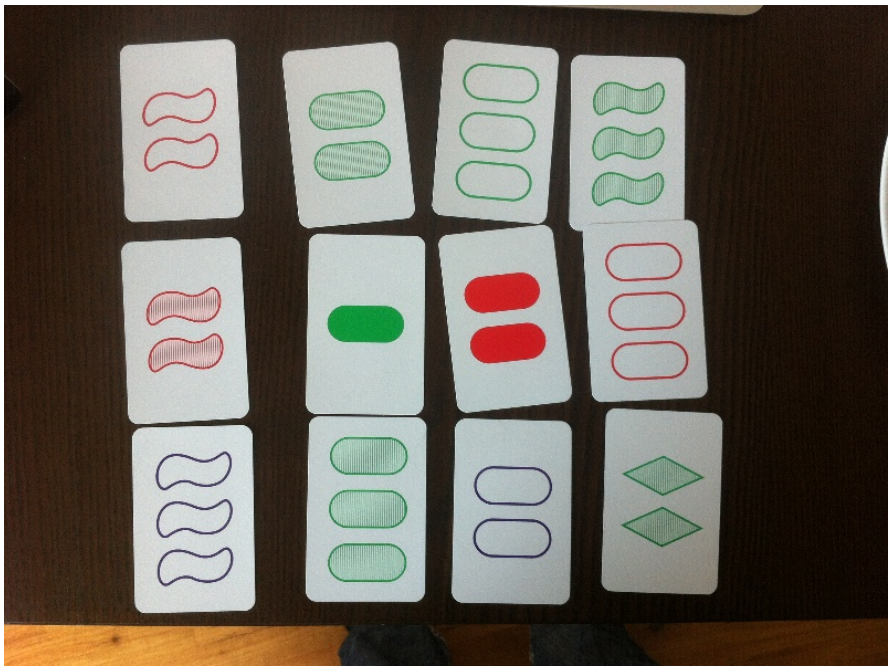


Figure : An image of 12 SET cards.

Method

Locating cards

Two options:

- **Top down approach** - using card borders.
- **Bottom up approach** - using the location of individual shapes.

Method

Locating cards

Two options:

- **Top down approach** - using card borders.
- **Bottom up approach** - using the location of individual shapes.

Classification of the cards

- **Shape** - comparison of shape contours
- **Fill and color** - classification models on features
- **Quantity** - follows from combining shapes

Method

Locating cards

Two options:

- **Top down approach** - using card borders.
- **Bottom up approach** - using the location of individual shapes.

Classification of the cards

- **Shape** - comparison of shape contours
- **Fill and color** - classification models on features
- **Quantity** - follows from combining shapes

Finding SETs

$O(n^2)$ algorithm

Method: Locating cards

Top down approach

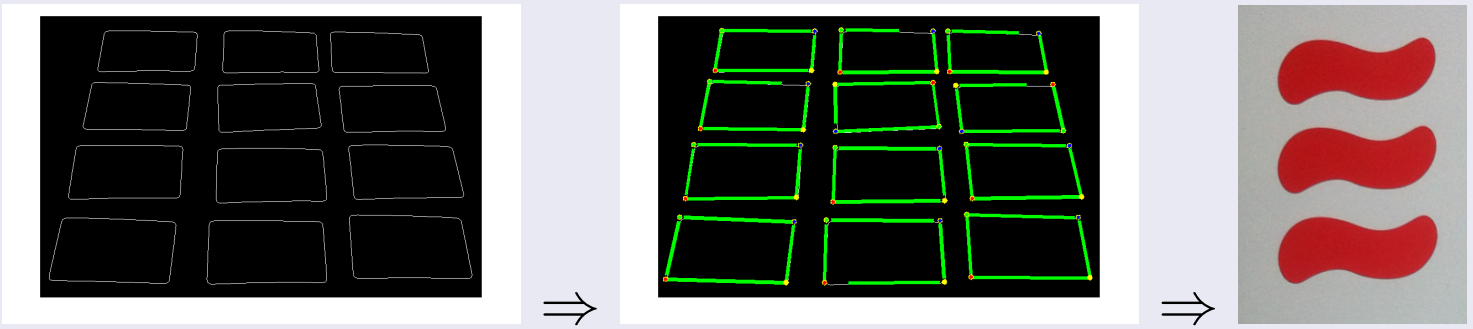
1. Finding contours of cards
2. Locating card corners
3. Extracting card image

Method: Locating cards

Top down approach

1. Finding contours of cards
2. Locating card corners
3. Extracting card image

Example

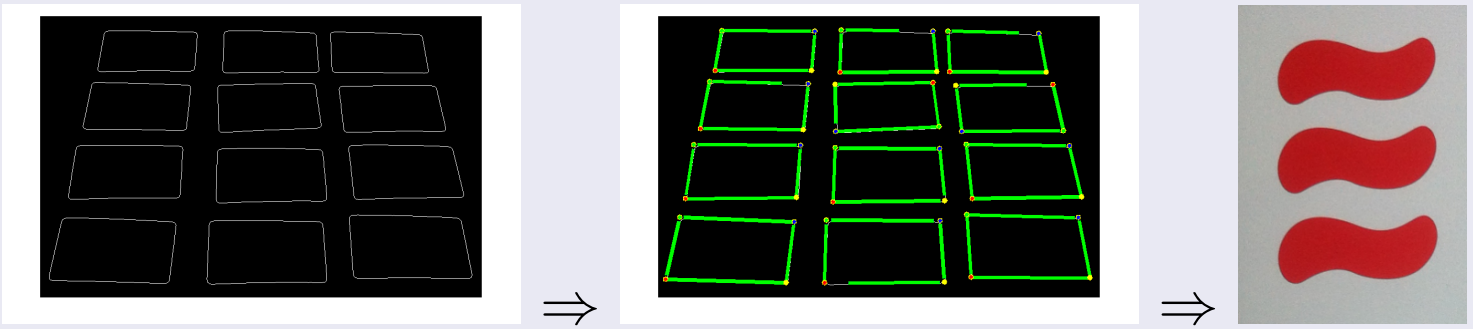


Method: Locating cards

Top down approach

1. Finding contours of cards
2. Locating card corners
3. Extracting card image

Example



Discussion

- Generally used method in literature
- Main disadvantage: cards may not overlap

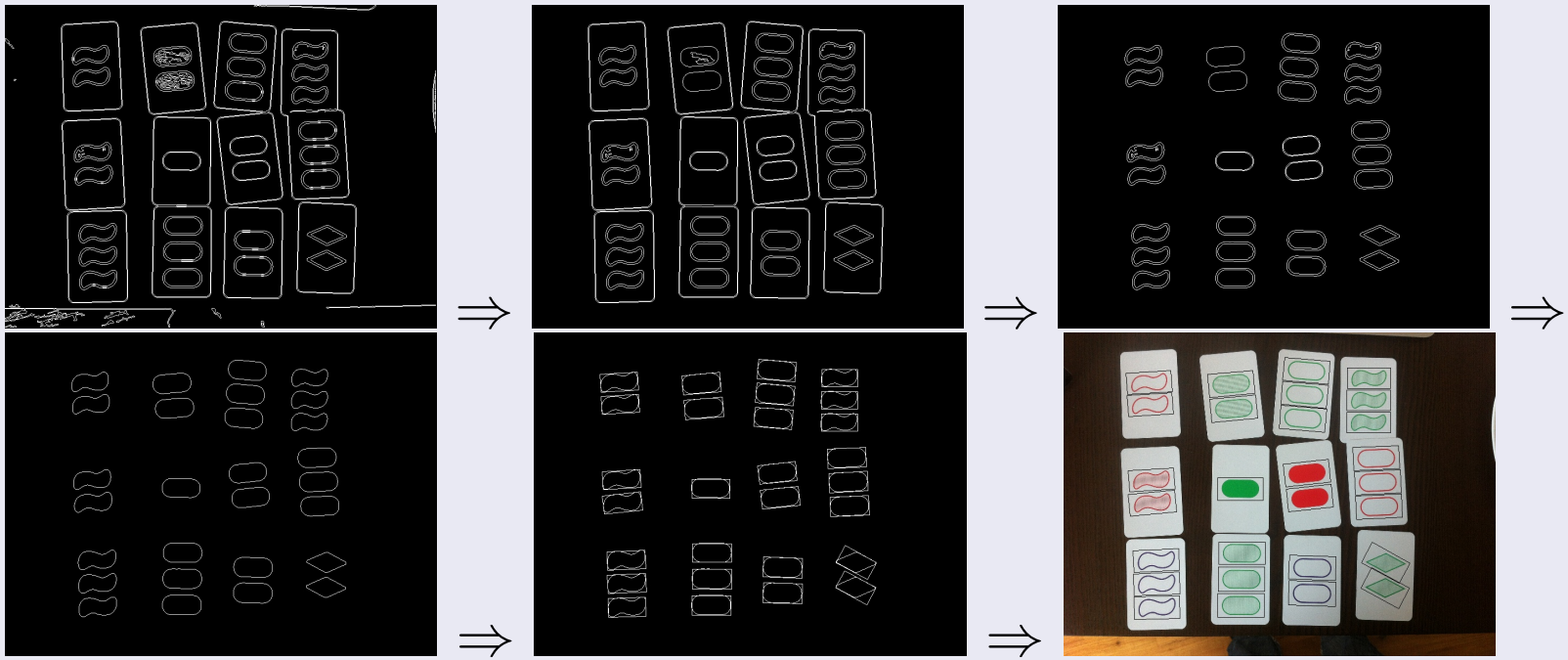
Method: Locating cards

Bottom up approach

1. Finding contours in the image
2. Filtering contours on topological features
3. Finding the group of shape contours
4. Extracting shape images
5. Combining shapes to cards

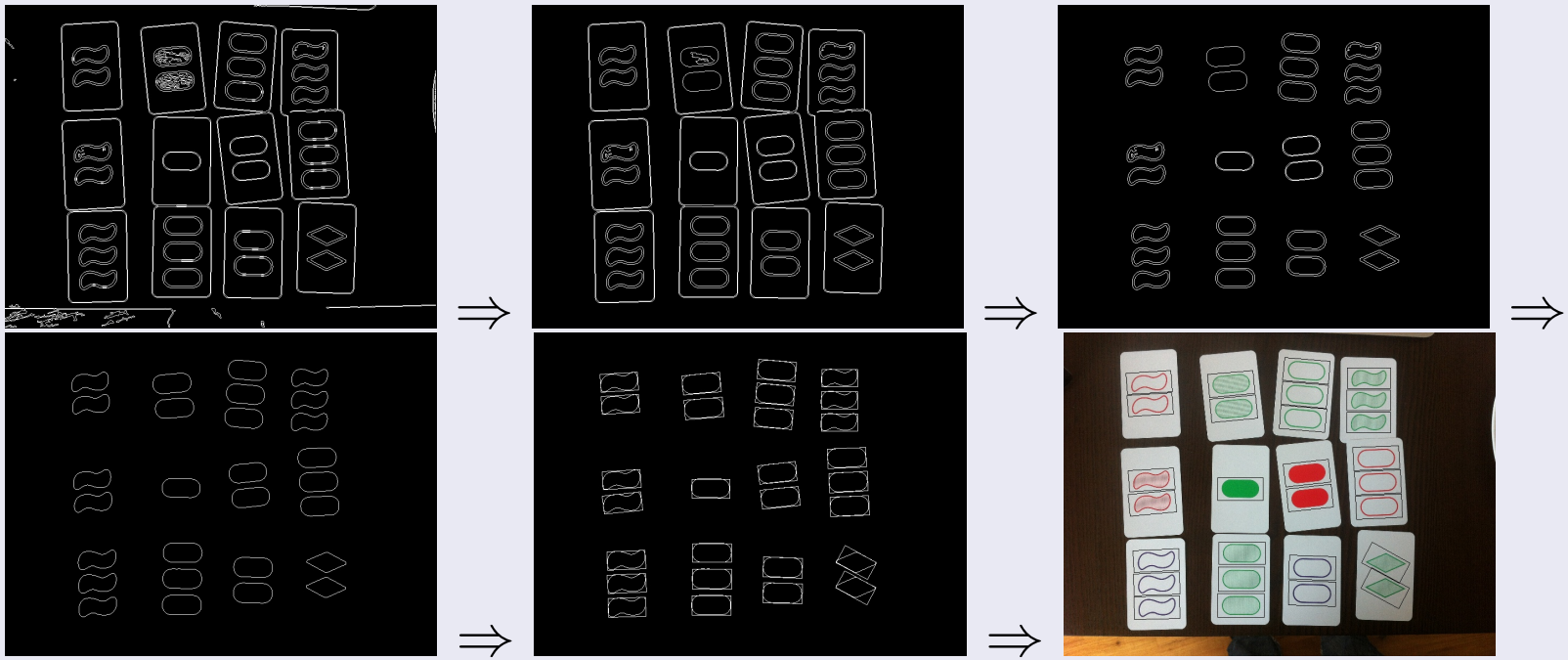
Method: Locating cards

Example



Method: Locating cards

Example



Discussion

- Method works well: classification based on this
- Combining to cards done after classification

Method: Classification of the cards

Shape

Two options:

- Using image moments
- Using absolute image difference

Method: Classification of the cards

Shape

Two options:

- Using image moments
- Using absolute image difference

Fill and color

- 7 features for fill, 11 for color
- Feature set reduction (PCA)
- 4 classification models

Method: Classification of the cards

Shape

Two options:

- Using image moments
- Using absolute image difference

Fill and color

- 7 features for fill, 11 for color
- Feature set reduction (PCA)
- 4 classification models

Quantity

Follows from combining shapes to cards

Method: Classification of the cards, the shape

Idea

- Reference shapes for each brand
- Comparison gives error
- Match shapes for brand with lowest average error

Method: Classification of the cards, the shape

Idea

- Reference shapes for each brand
- Comparison gives error
- Match shapes for brand with lowest average error

Using image moments

$$I(A,B) = \sum_{i=1}^{7} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right| \quad (1)$$

$m_i^A = sign(h_i^A) \cdot \log h_i^A$ and $m_i^B = sign(h_i^B) \cdot \log h_i^B$, h_i^A and h_i^B are the Hu-moments

Method: Classification of the cards, the shape

Idea

- Reference shapes for each brand
- Comparison gives error
- Match shapes for brand with lowest average error

Using image moments

$$I(A,B) = \sum_{i=1}^{7} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right| \quad (1)$$

$m_i^A = sign(h_i^A) \cdot \log h_i^A$ and $m_i^B = sign(h_i^B) \cdot \log h_i^B$, h_i^A and h_i^B are the Hu-moments

Using absolute image difference

Lowest error:  $\Rightarrow$  or  or  .

Method: Classification of the cards, the fill

Preprocessing

- Convert to grayscale
- Maximize contrast
- Extract region of interest

Method: Classification of the cards, the fill

Calculation of features

- **Mean luminance** - the mean of the pixel intensities.
- **Quantiles of the luminance distribution** - the 5%, 25%, 75% and 95%-quantiles of the distribution of the pixel intensities.
- **Mean absolute Laplacian** - the mean of the values obtained from applying the Laplacian operator on the matrix with pixel intensities.
- **Mean absolute difference to blurred image** - the mean of the absolute differences in pixel intensities between the image and a heavily blurred version of the image.

Method: Classification of the cards, the fill

Feature set reduction

- Find low dimensional representation of data
- Prevents overfitting in high dimensional space
- Relevant information should not be lost

Method: Classification of the cards, the fill

Feature set reduction

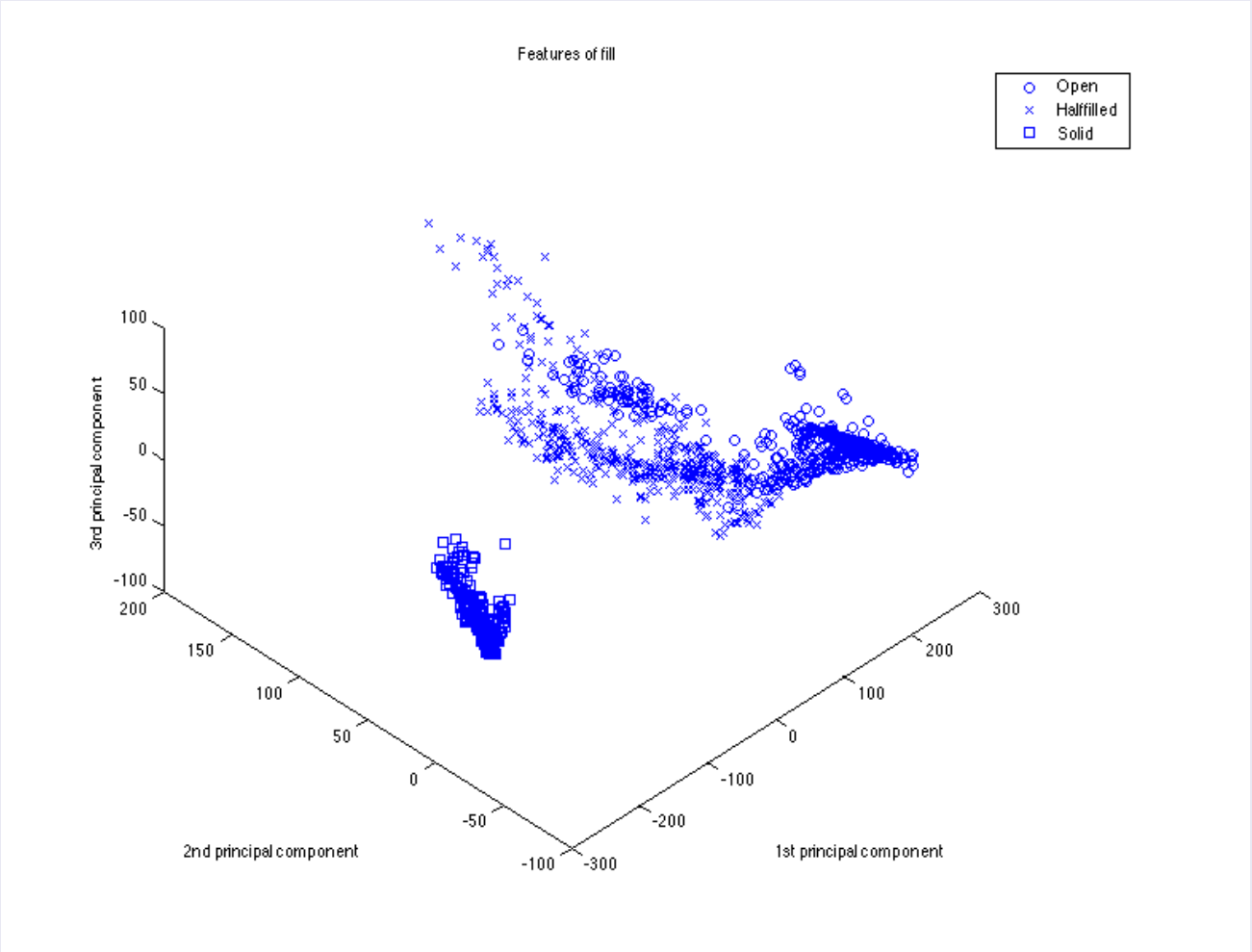
- Find low dimensional representation of data
- Prevents overfitting in high dimensional space
- Relevant information should not be lost

Principle component analysis

- Transformation of feature space
- First components represent most variance
- Better than feature subset selection

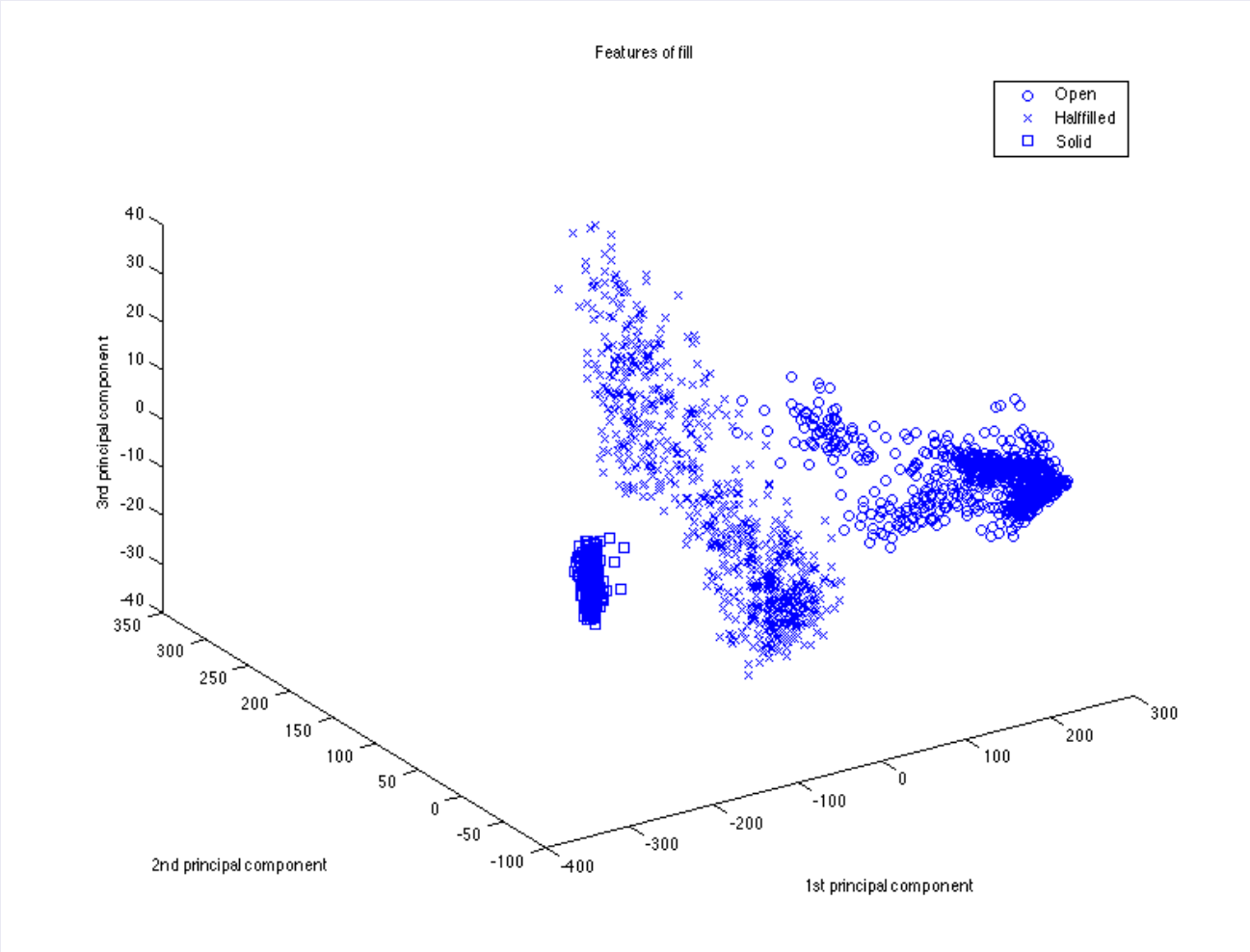
Method: Classification of the cards, the fill

3 principle components, NNN Games



Method: Classification of the cards, the fill

3 principle components, Ravensburger



Method: Classification of the cards, the fill

Training of a classification model

- Training on a training set (discussed later)
- Training on features resulting from PCA

Method: Classification of the cards, the fill

Training of a classification model

- Training on a training set (discussed later)
- Training on features resulting from PCA

Classification models

- Support Vector Machines
- K-Nearest Neighbor
- Normal Bayes Classifier
- Gradient Boosted Trees

Method: Classification of the cards, the color

,

Preprocessing

- Convert to colorspace: YCrCb
- Selecting pixels of interest

Method: Classification of the cards, the color

,

Preprocessing

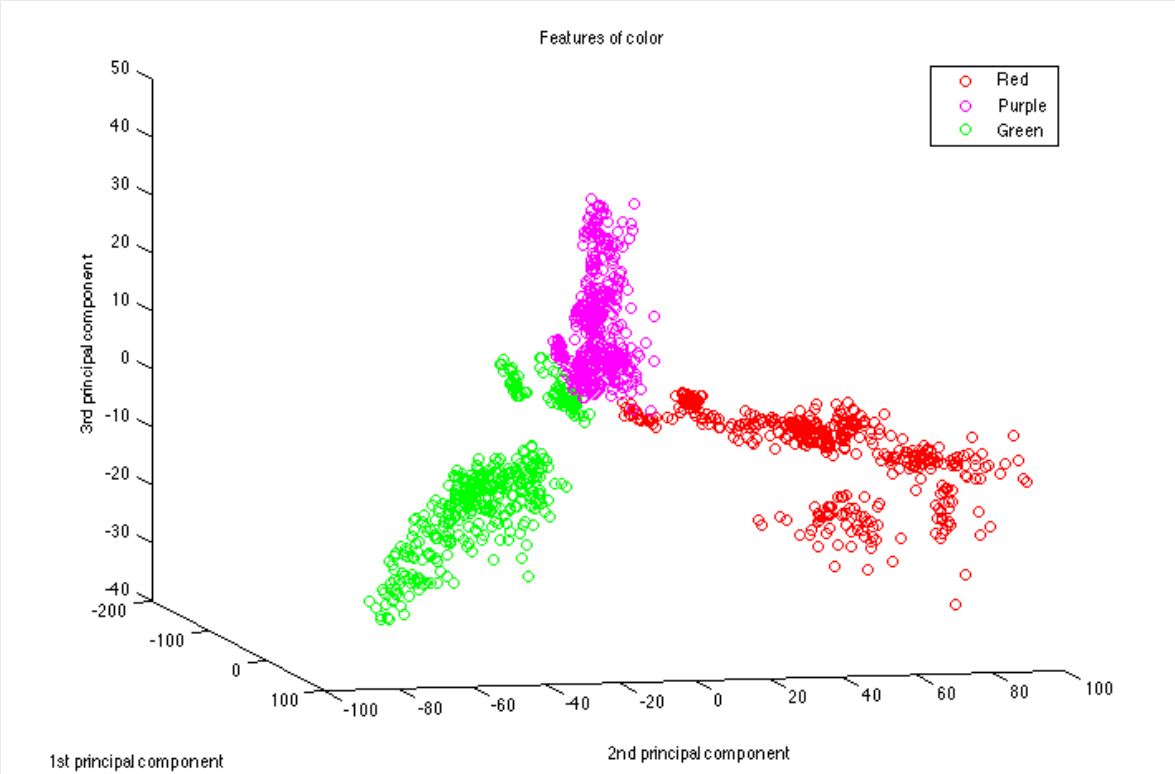
- Convert to colorspace: YCrCb
- Selecting pixels of interest

Calculation of features

- **Mean Cr and Cb** - the means of the Cr and Cb values of the pixels of interest.
- **Quantiles of the Cr and Cb distribution** - the 5%, 25%, 75% and 95%-quantiles of the distributions of the Cr and Cb values of the pixels of interest.
- **Lightness** - the 95%-quantile of the distribution of the Y values of the *full* image.

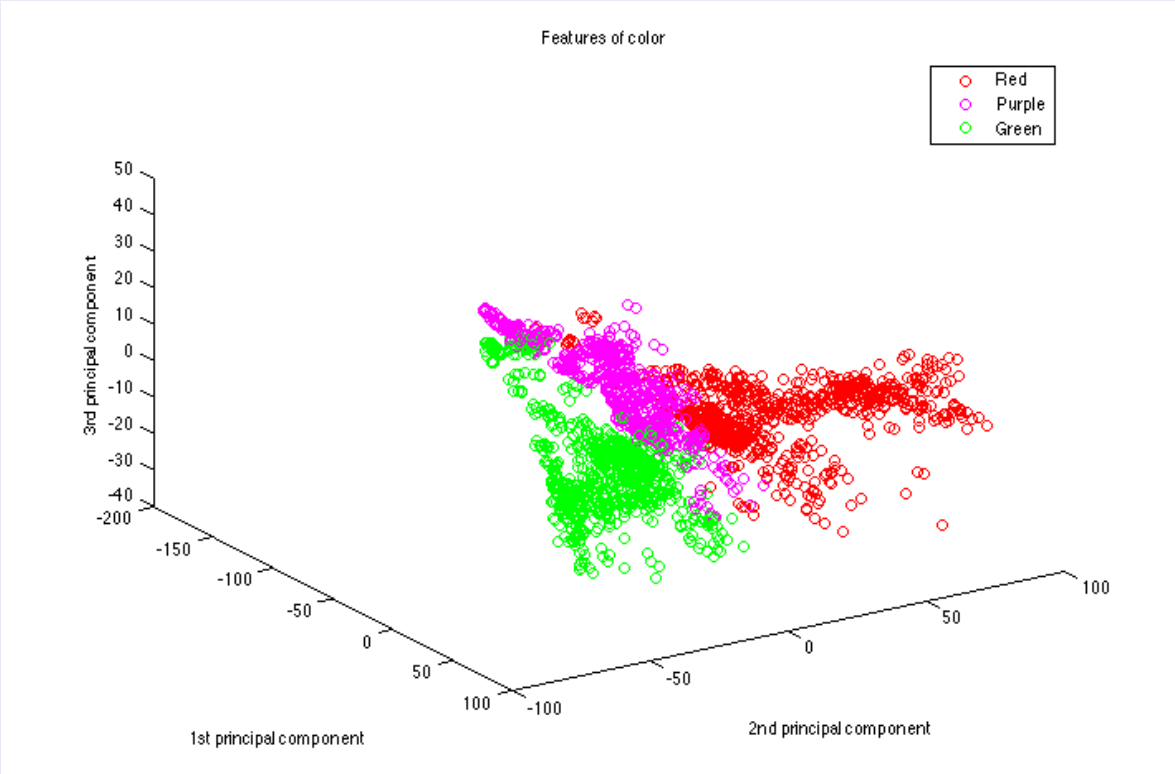
Method: Classification of the cards, the color

3 principle components, NNN Games



Method: Classification of the cards, the color

3 principle components, Ravensburger



Method: Classification of the cards, the quantity

Combining shapes

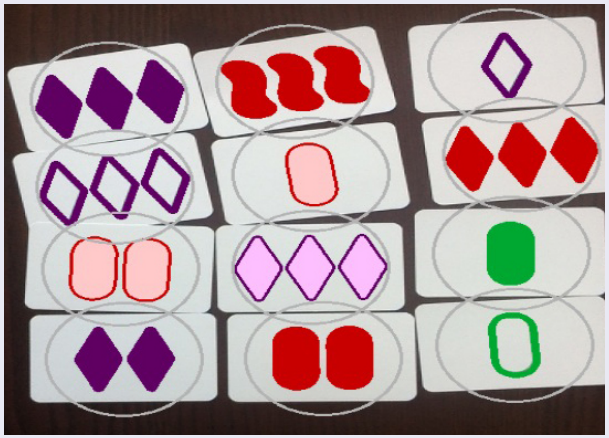
- Equal shape, fill and color
- Distance smaller than average width
- Recursion to label shapes
- Results in quantity and cards

Method: Classification of the cards, the quantity

Combining shapes

- Equal shape, fill and color
- Distance smaller than average width
- Recursion to label shapes
- Results in quantity and cards

Example



Method: Finding SETs

Idea

- Each pair forms SET with exactly one card
- For each pair, check if third card available

Method: Finding SETs

Idea

- Each pair forms SET with exactly one card
- For each pair, check if third card available

Implementation

- Build index with cards
- Iterate over pairs
- Property of third card for each pair:
  - Property equal $\Rightarrow$ third card equal
  - Property different $\Rightarrow$ third card remaining property
- Check if third card available and SET not already found

Results

Training set

- 2266 shape images for Ravensburger
- 1299 shape images for NNN Games

Results

Training set

- 2266 shape images for Ravensburger
- 1299 shape images for NNN Games

Experiment setup

- Shapes: static method using single reference shape
- Fill and color: 5-fold crossvalidation

Results

Shape

- Absolute image difference works best

Results

Shape

- Absolute image difference works best

Color and fill

- Generally more principle components result in higher accuracy
- SVM generally performs best

Results

Shape

- Absolute image difference works best

Color and fill

- Generally more principle components result in higher accuracy
- SVM generally performs best

Combined accuracy

Using SVM on all principle components

Edition	Color	Fill	Shape	All
Ravensburger	96.03 %	99.87 %	99.74 %	95.72 %
NNN Games	99.77 %	99.46 %	99.92 %	99.15 %

Implementation

The SET app

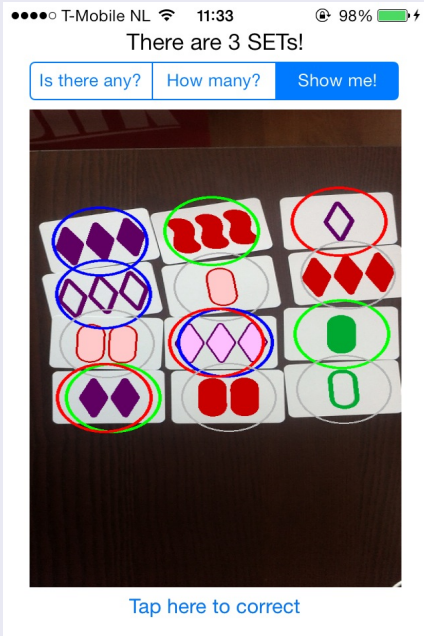
- Implementation in an iOS app
- Using OpenCV framework
- Corrective mechanism for errors

Implementation

The SET app

- Implementation in an iOS app
- Using OpenCV framework
- Corrective mechanism for errors

Screenshots



Discussion

Accuracy

- High accuracy for individual shape
- All 12 cards correct roughly with probability $0.957^{12 \cdot 2} = 0.350$ for RVB and $0.992^{12 \cdot 2} = 0.815$ for NNN

Discussion

Accuracy

- High accuracy for individual shape
- All 12 cards correct roughly with probability $0.957^{12 \cdot 2} = 0.350$ for RVB and $0.992^{12 \cdot 2} = 0.815$ for NNN

In practice

- Indeed frequently around one shape incorrect
- Feedback mechanism required
- Occasionally 'clear' image incorrect so training set could be enlarged

Discussion

Accuracy

- High accuracy for individual shape
- All 12 cards correct roughly with probability $0.957^{12 \cdot 2} = 0.350$ for RVB and $0.992^{12 \cdot 2} = 0.815$ for NNN

In practice

- Indeed frequently around one shape incorrect
- Feedback mechanism required
- Occasionally 'clear' image incorrect so training set could be enlarged

Future work

- Use of more texture descriptors from literature
- Use of color image normalization methods

The end

Did you find these SETs?