```
clear all
addpath('./functions');
addpath('./Images_mandatory');
addpath('./Images_advanced');
warning off
```

Project: Memory

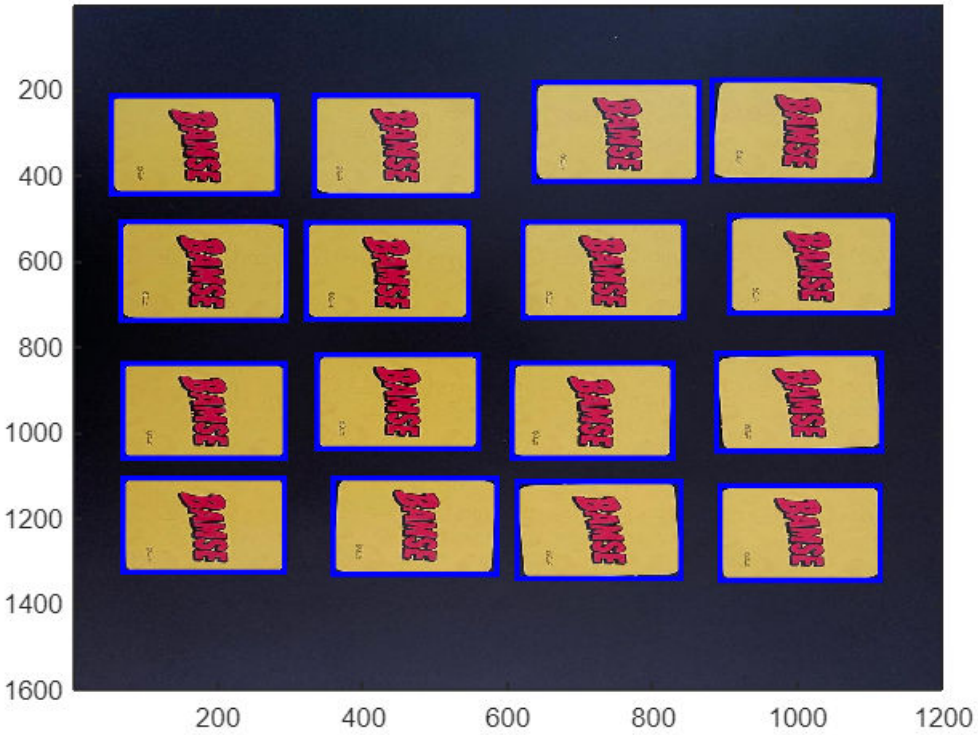
The first step is to load the images containing the memory cards, this is done using the function load_images. Which loads the images and reads them.

```
images = load_images();
```

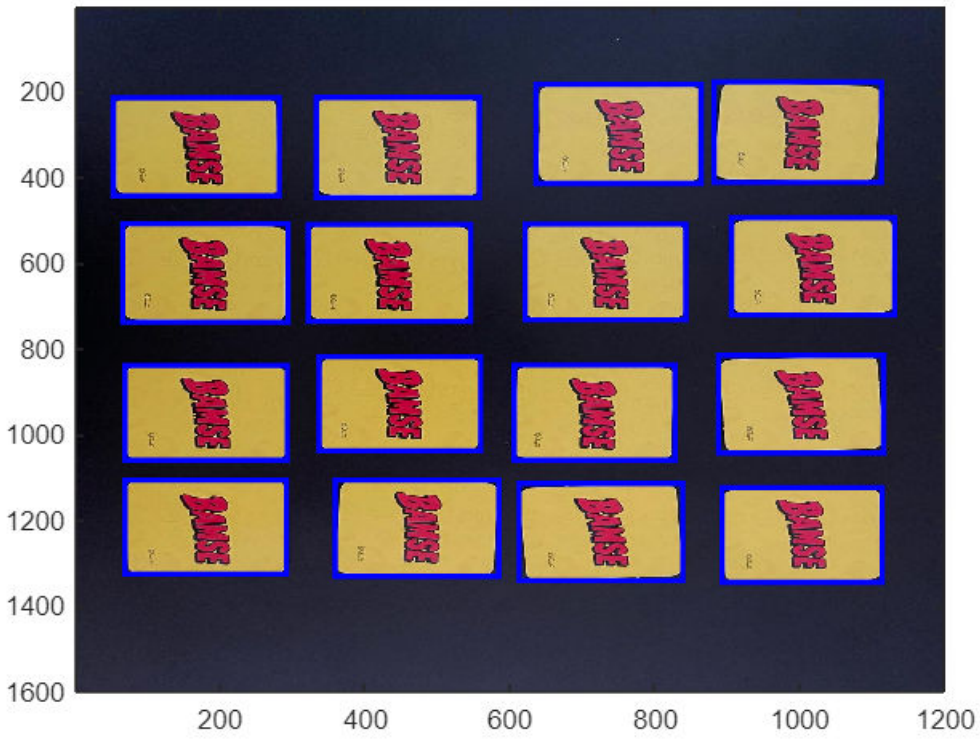
Game set up

At the beginning of the game, all of the cards are face down and bounding boxes are drawn around the cards.

```
img_n = 1;                          % Start with the first image
image = images{img_n};
binary_im = binarize_image(image);  % Make a binary image of the first image
[rec] = create_boundary_box(binary_im,image);    % Create boundary boxes
```



```
figure;
imagesc(image);
hold on
for i = 1:length(rec)
    pos = rec(i).Position;
    rectangle('Position',pos,'EdgeColor','b','LineWidth',2);
end
```




```
disp('Player 1, please choose two cards to turn')
```

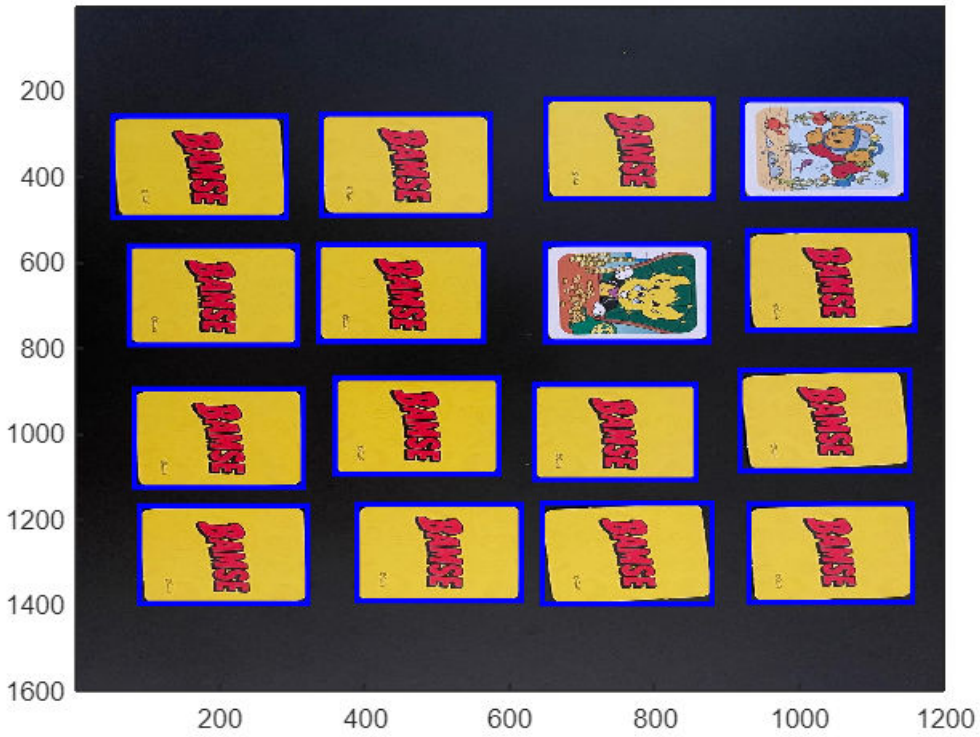
```
Player 1, please choose two cards to turn
```

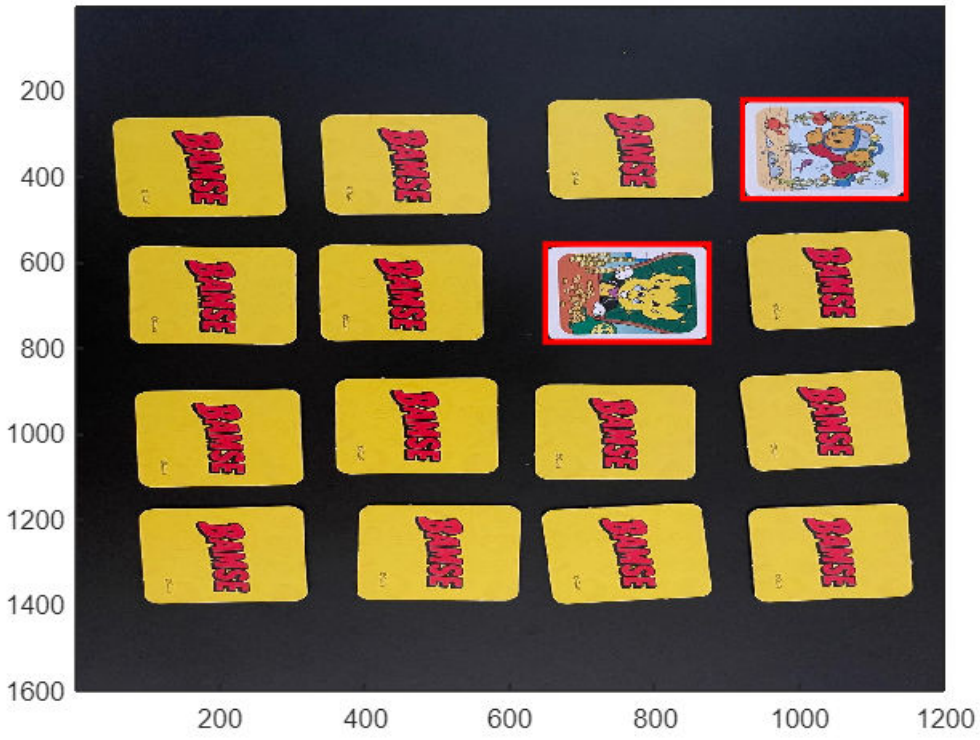
Playing the game

Once the first image of the back of all of the cards has been displayed, player 1 is asked to turn two cards around. If the cards are a match, player 1 can continue, otherwise player 2 can play.

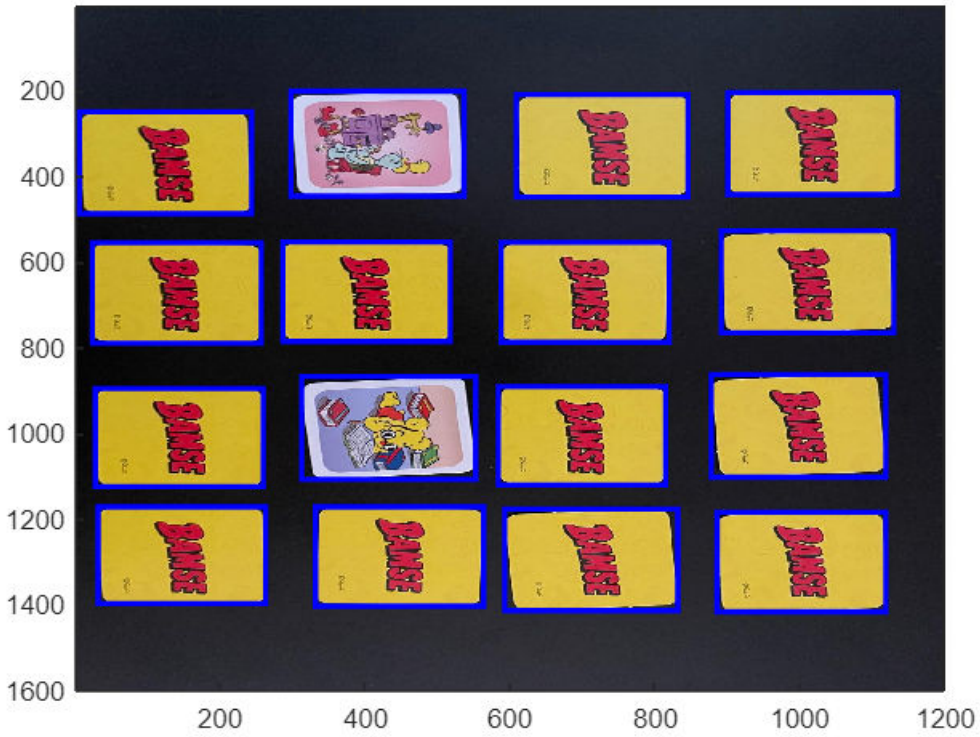
```
% Initializing the current player and the scores of player one and 2
current_player = 1;
score_1 = 0;
score_2 = 0;

for n = 2:length(images)
    box = [];
```

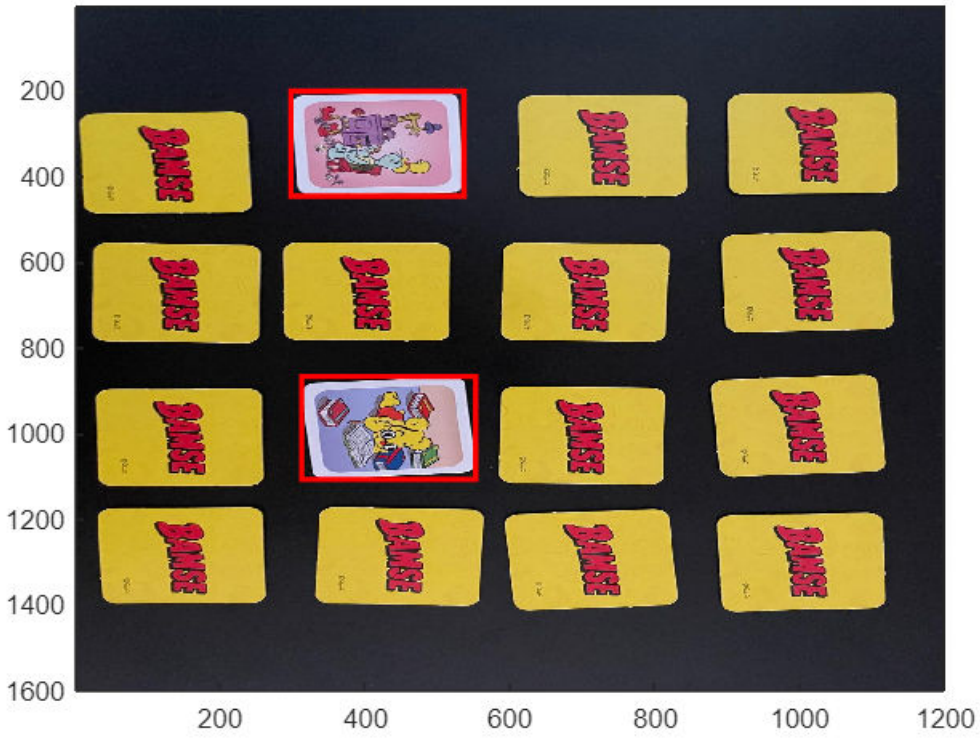
```
    card_up = [];
    card_down = [];

    image = images{n};


    binary_im = binarize_image(image);  % Make a binary image of the first image
    box = create_boundary_box(binary_im,image);     % Create all of the boundary
boxes

    % See which cards are face up
    [card_up,card_down] = card_up_or_down(box, image);

    % See from the face up cards if they are matching
    match = matching_cards(card_up,image);

    % if there is a match, increase the score
    if match == 1
        [score_1,score_2] = players_score(current_player,score_1,score_2);

        figure;
        imagesc(image)
        hold on
        for r = 1:2
            pos = card_up(r).Position;
            rectangle('Position',pos,'EdgeColor','g','LineWidth',2);
        end
        hold off
    else
        figure;
        imagesc(image)
        hold on
        for r = 1:2
            pos = card_up(r).Position;
            rectangle('Position',pos,'EdgeColor','r','LineWidth',2);
        end
        hold off
    end

    % Determine whether the current player is switched or not
    current_player = player_turn(match, current_player);


end
```



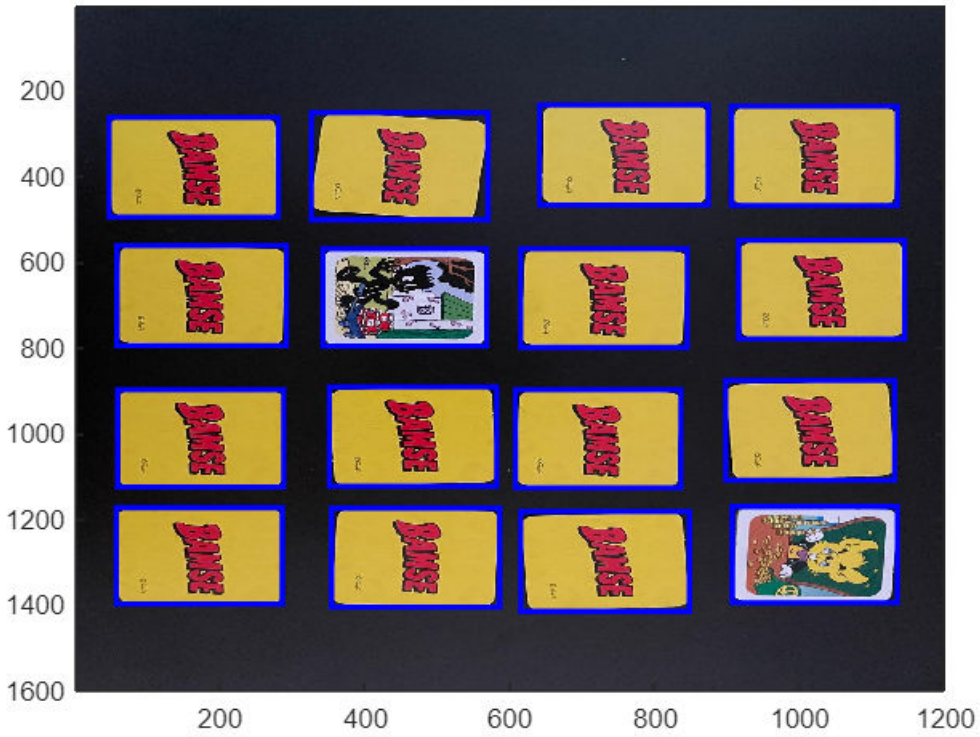


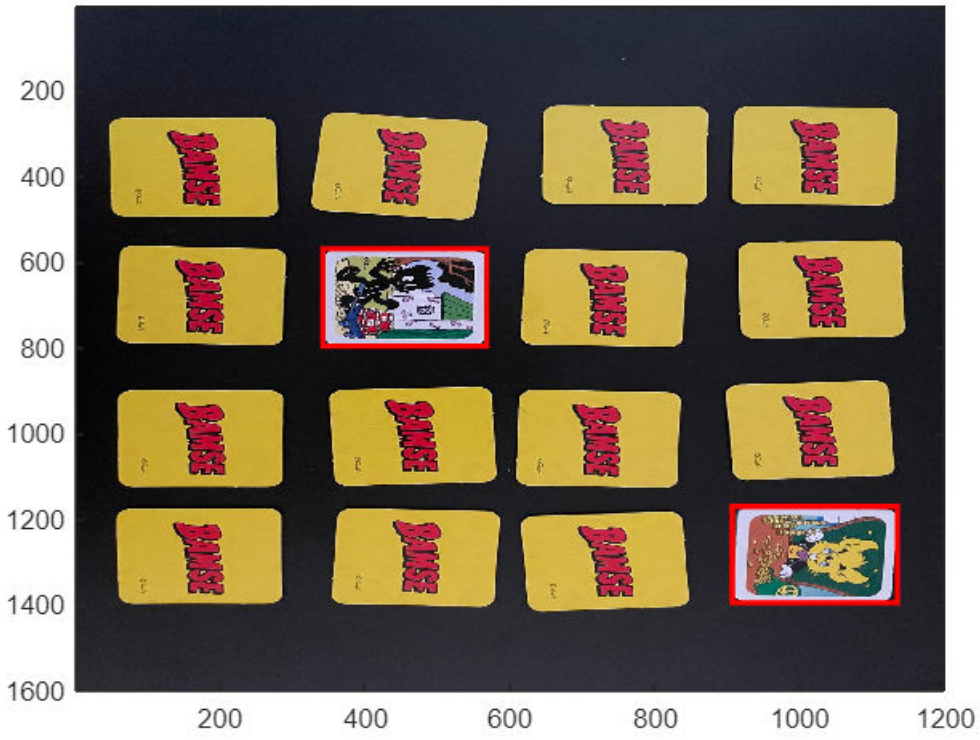
No match! Turn the two cards face down again. Player 2 may then continue



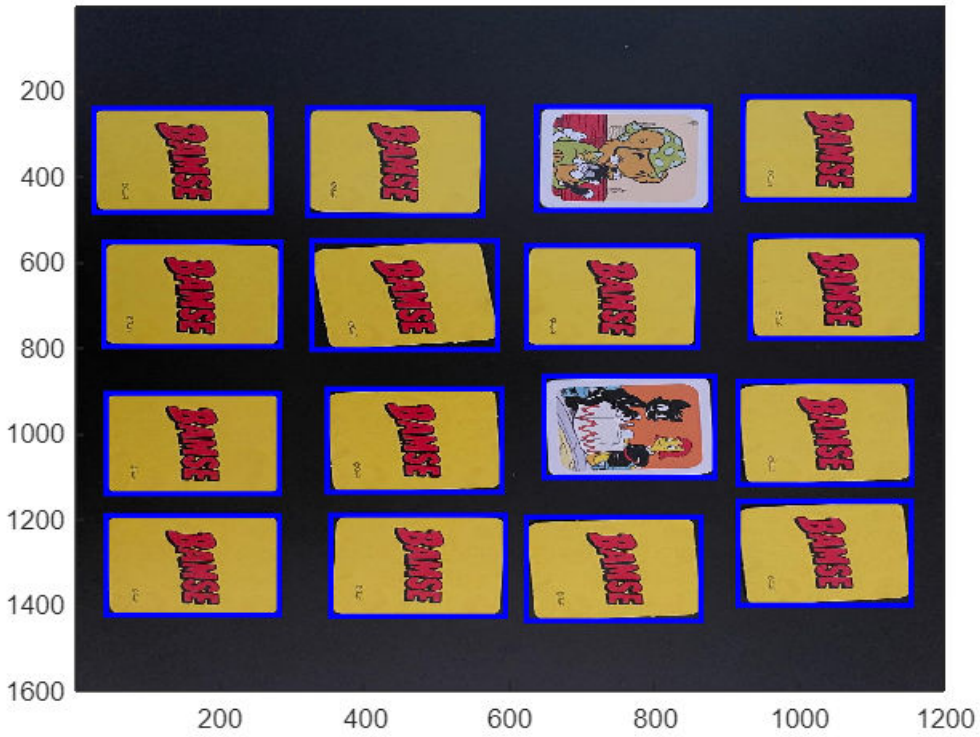



```
No match! Turn the two cards face down again. Player 1 may then continue
```

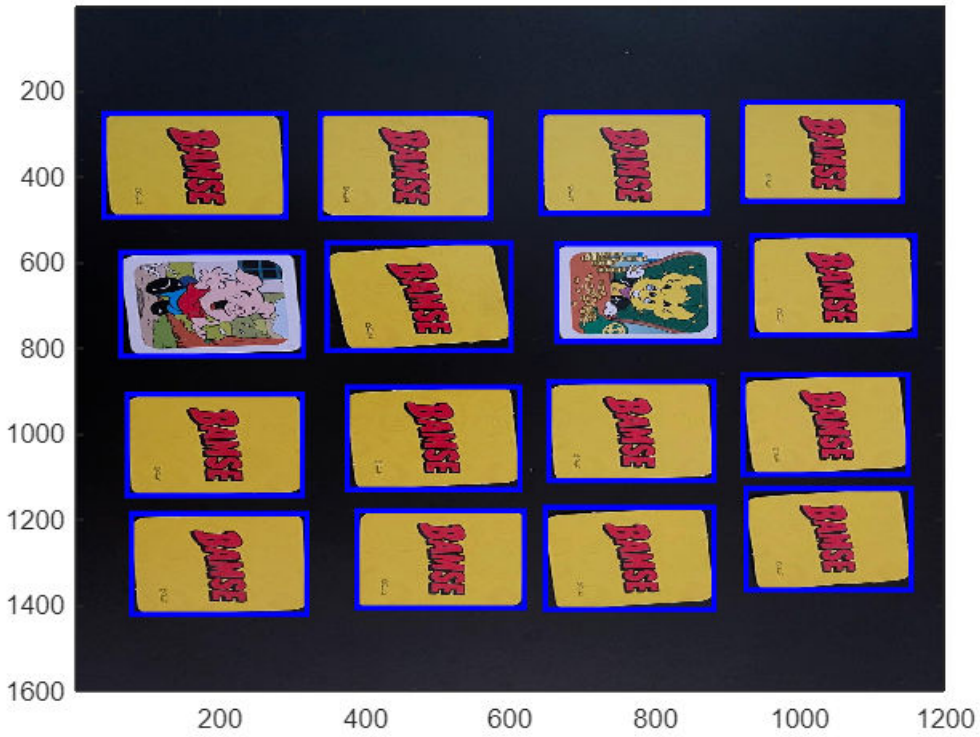


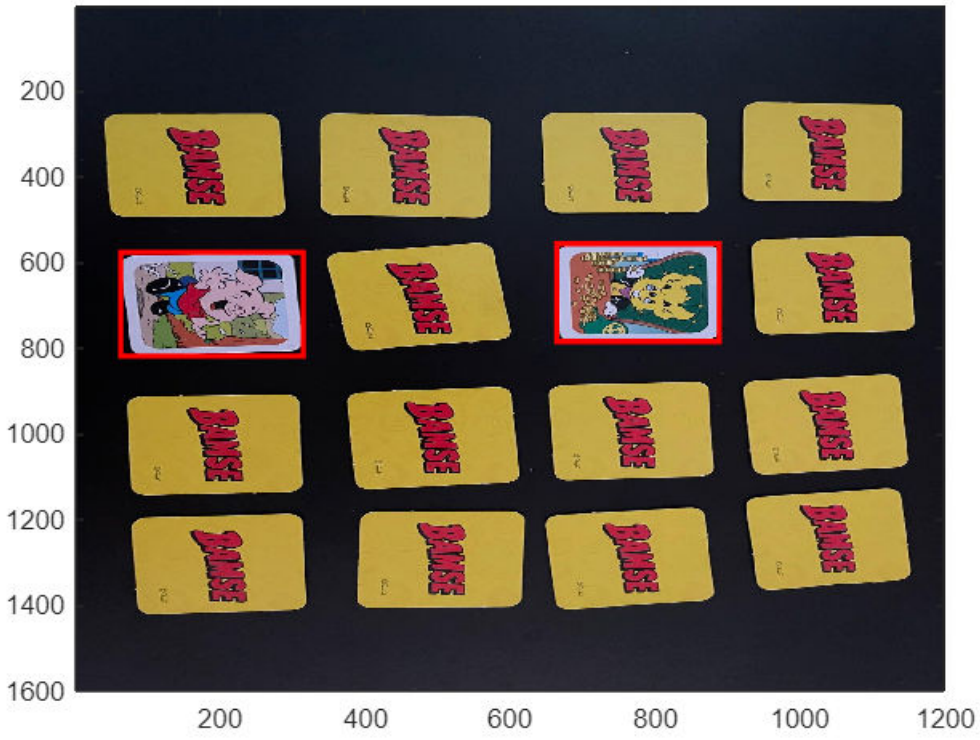


No match! Turn the two cards face down again. Player 2 may then continue

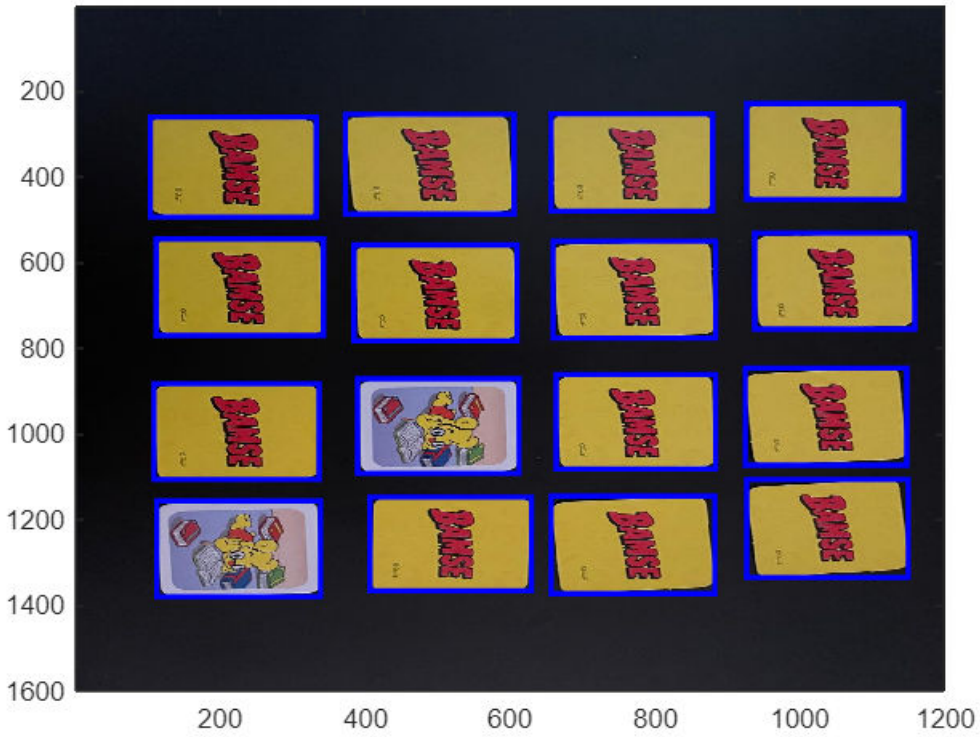


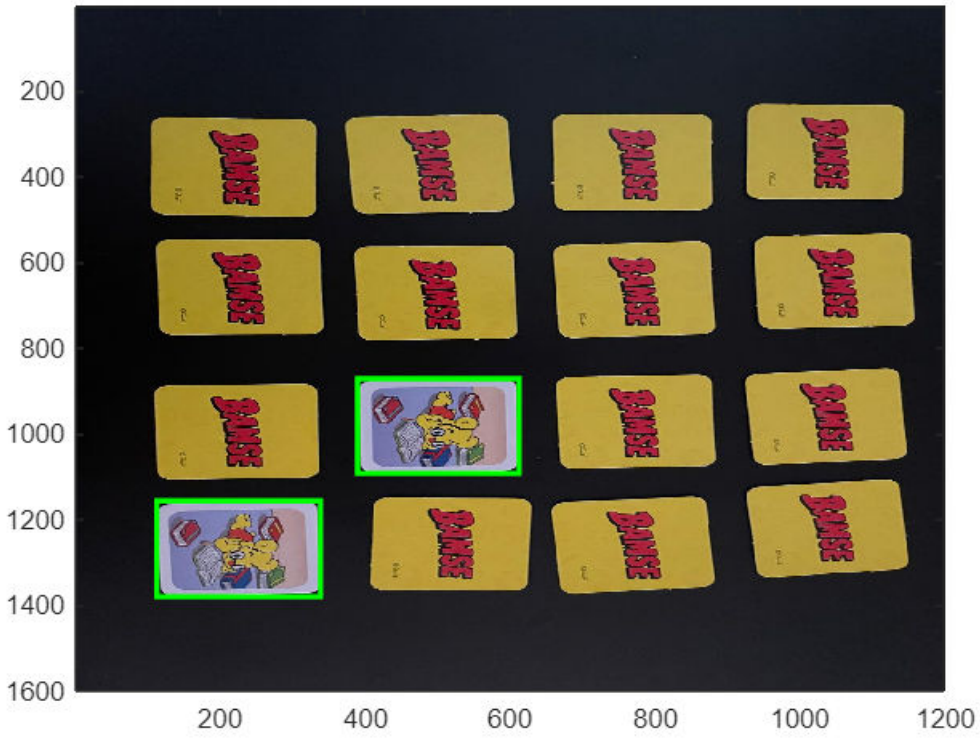
No match! Turn the two cards face down again. Player 1 may then continue



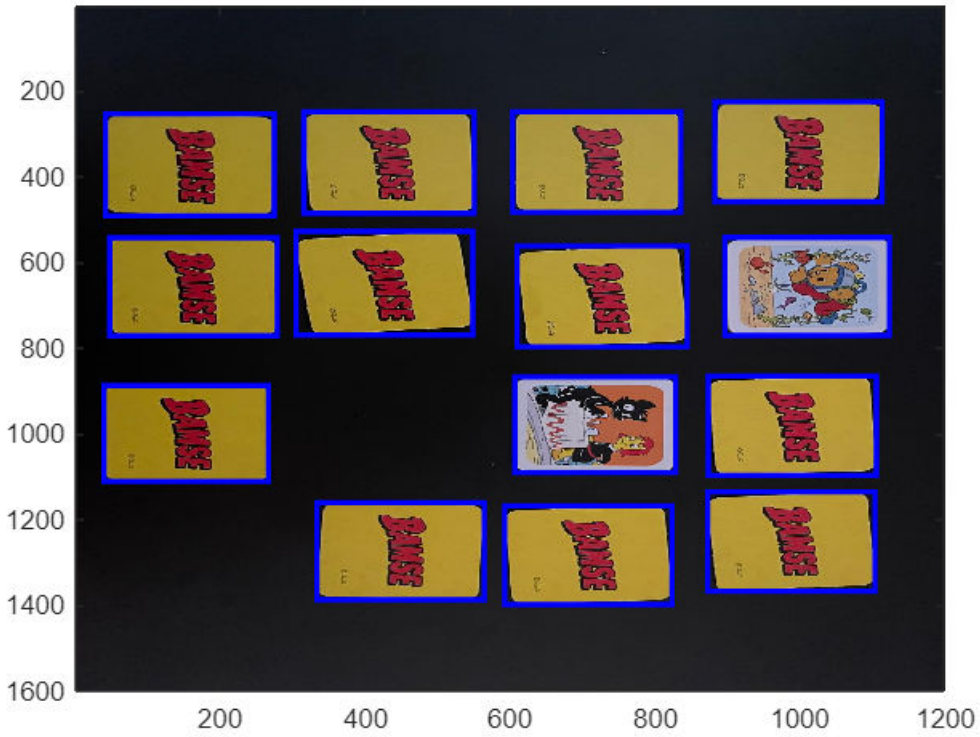


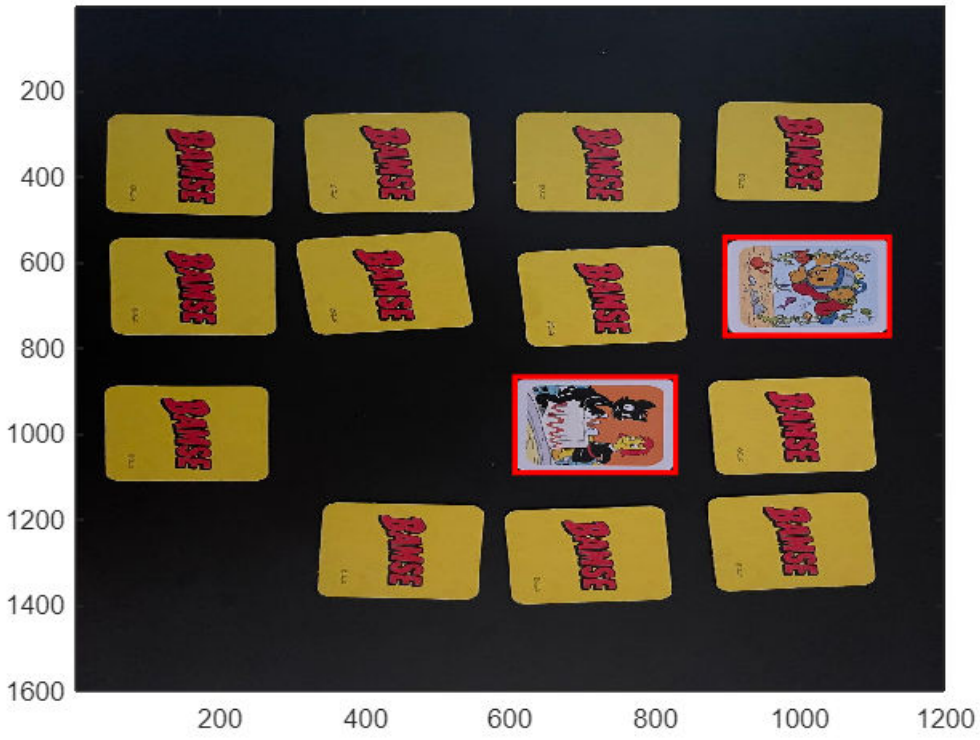
No match! Turn the two cards face down again. Player 2 may then continue



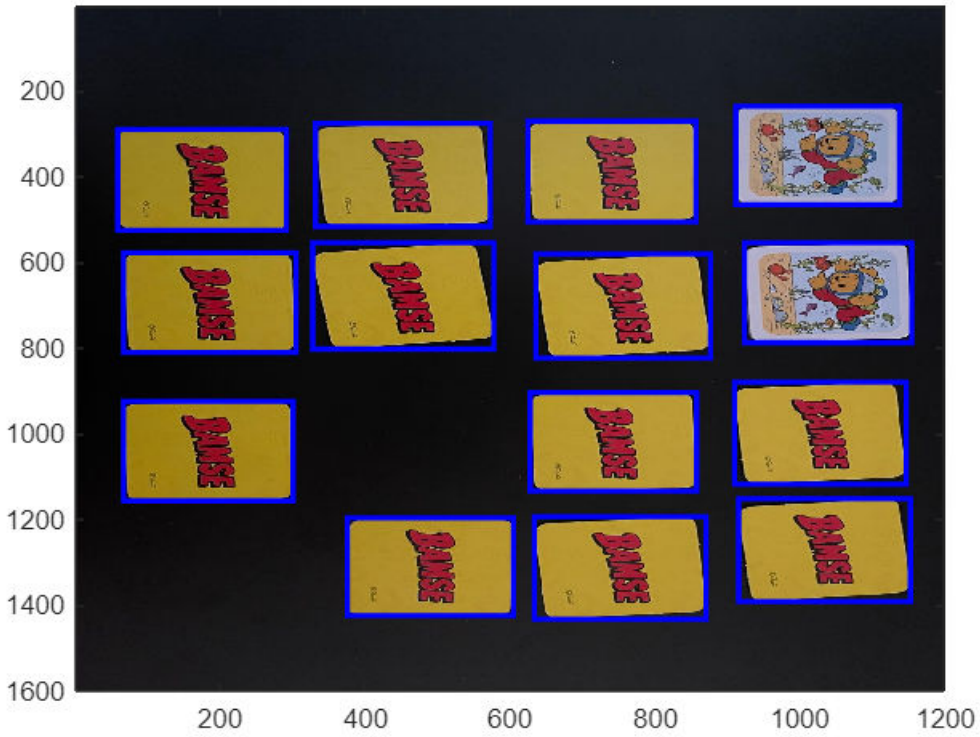


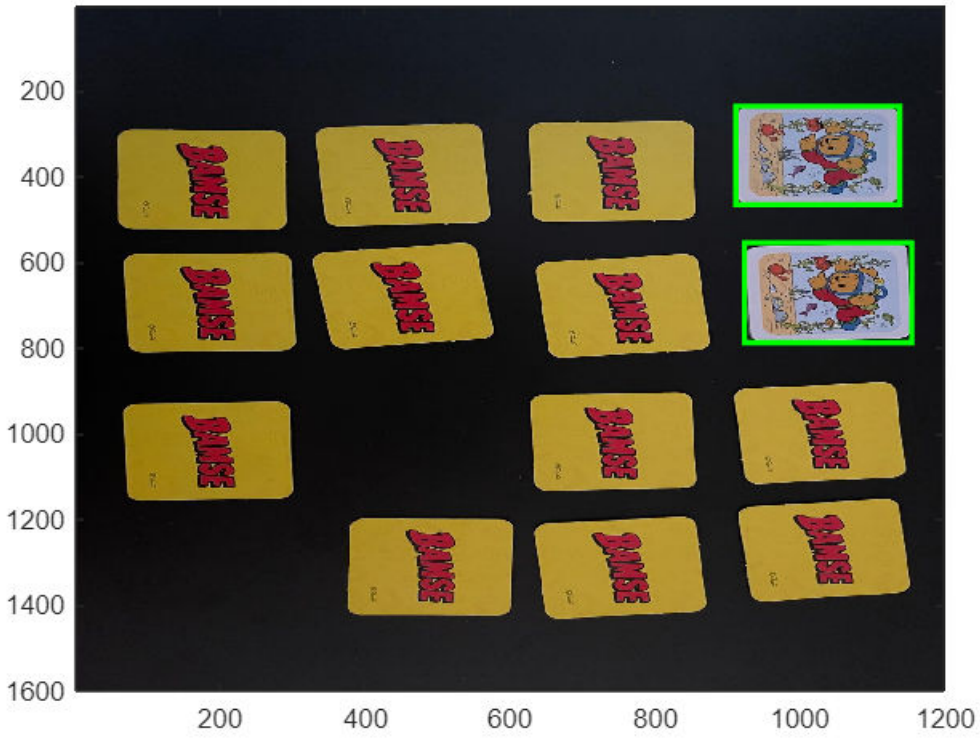
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



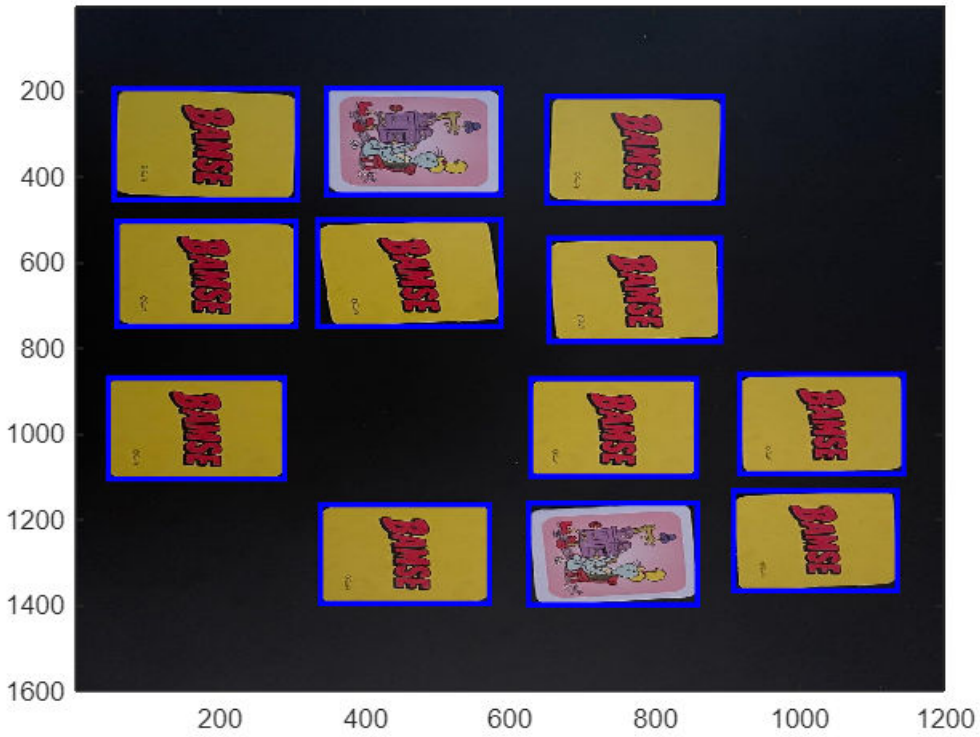


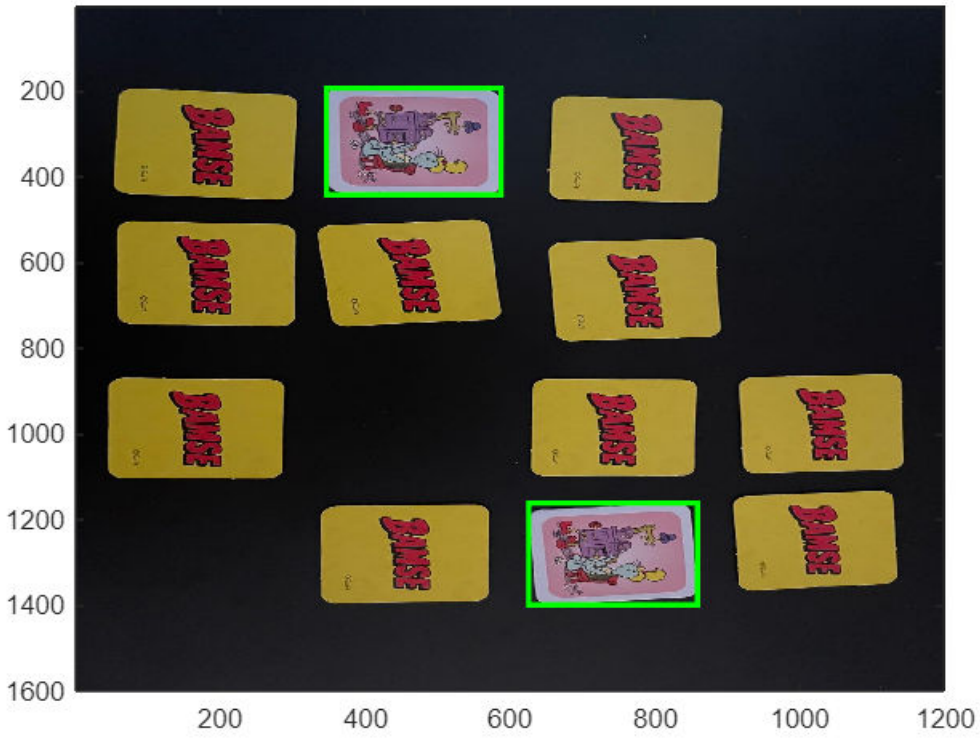
No match! Turn the two cards face down again. Player 1 may then continue



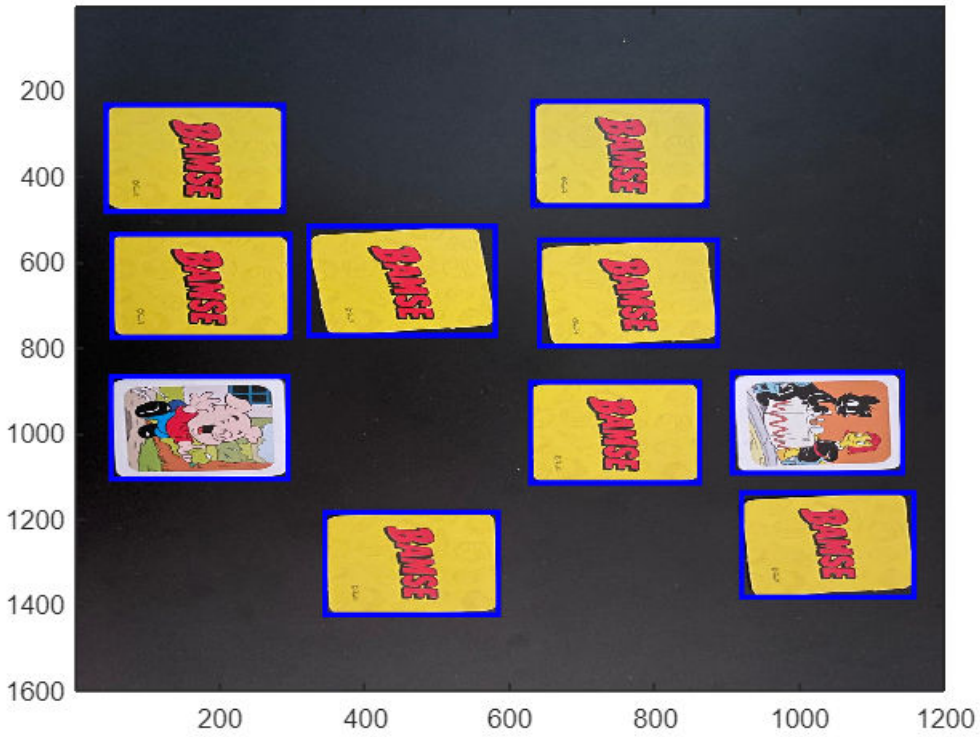


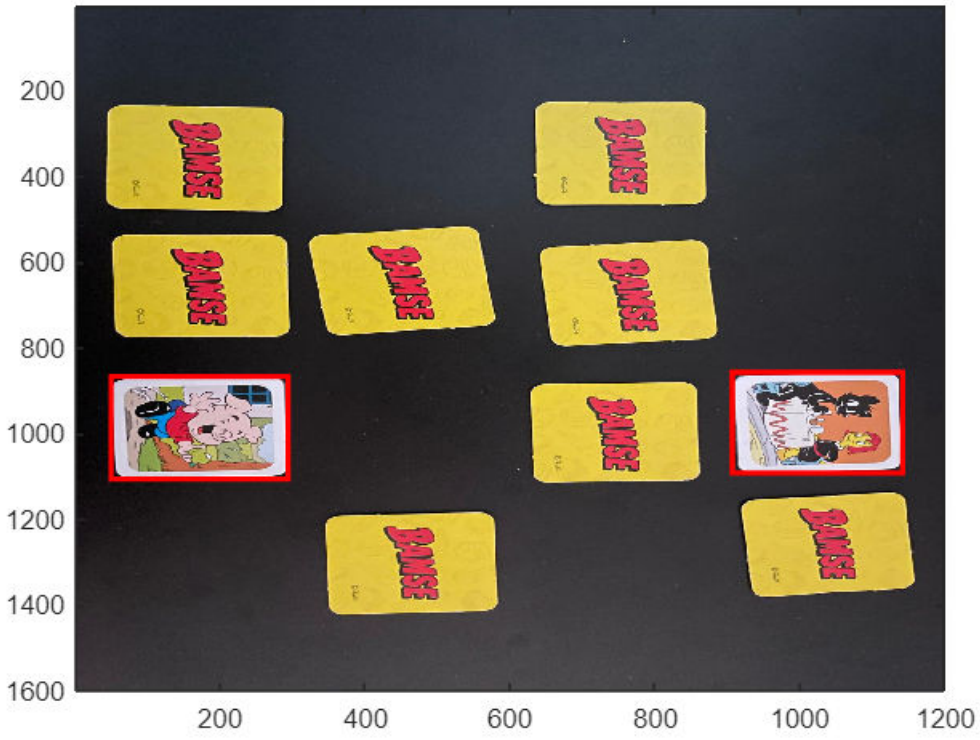
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



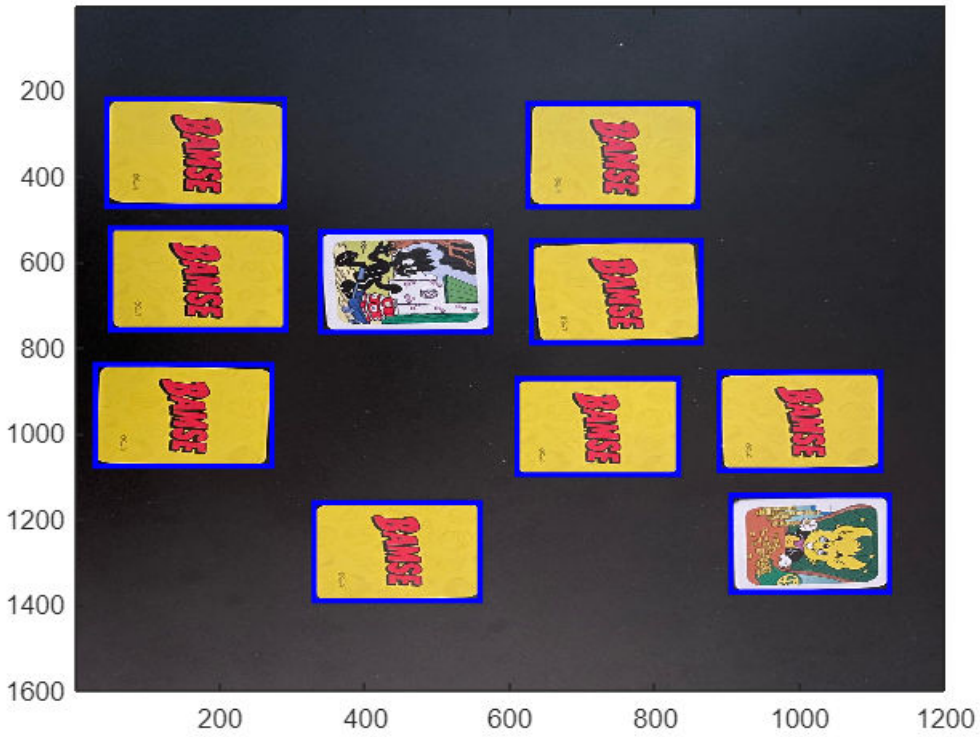


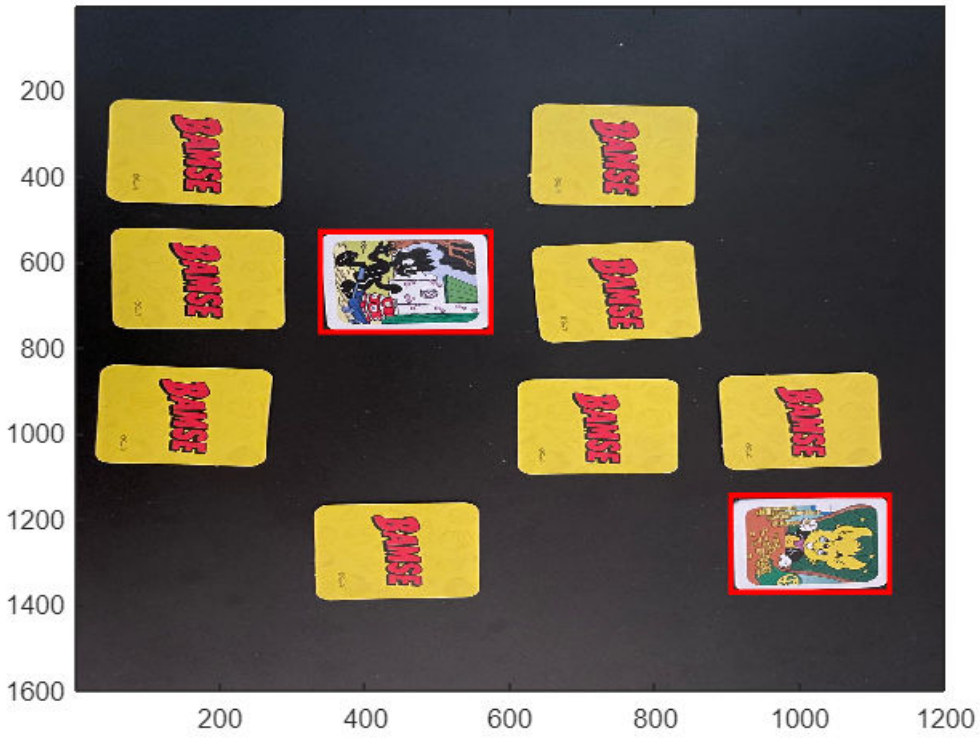
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



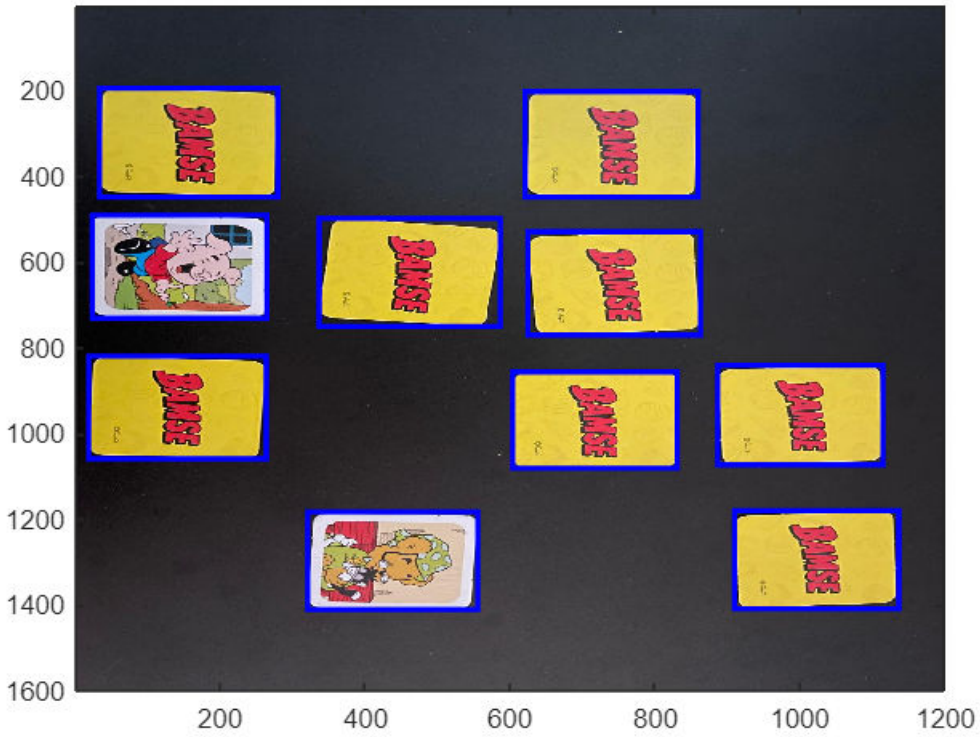


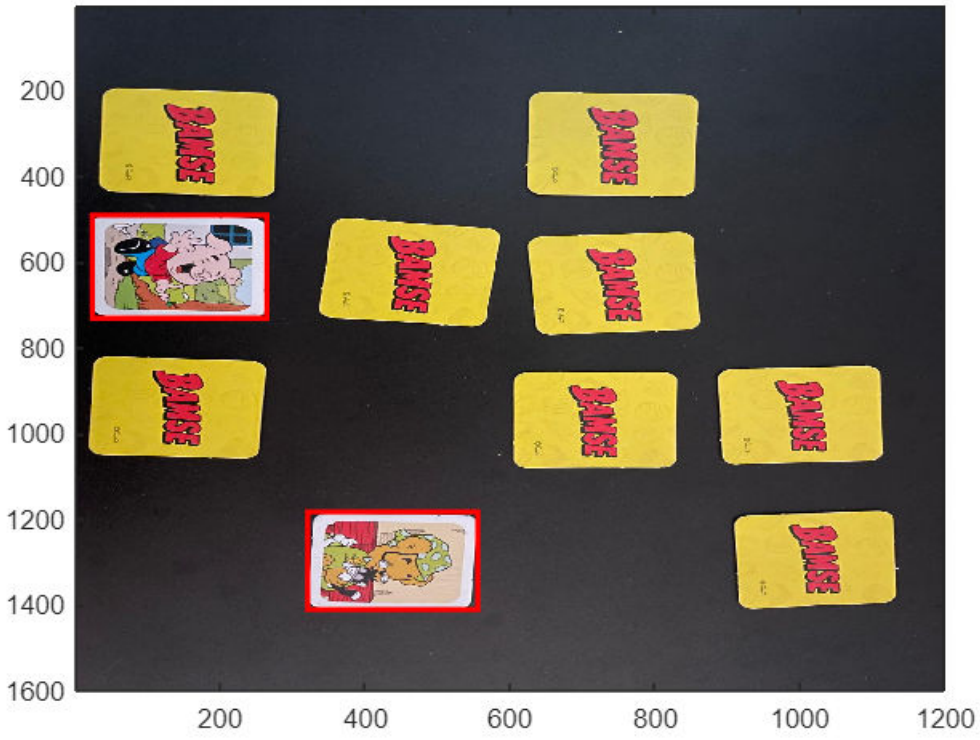
No match! Turn the two cards face down again. Player 2 may then continue



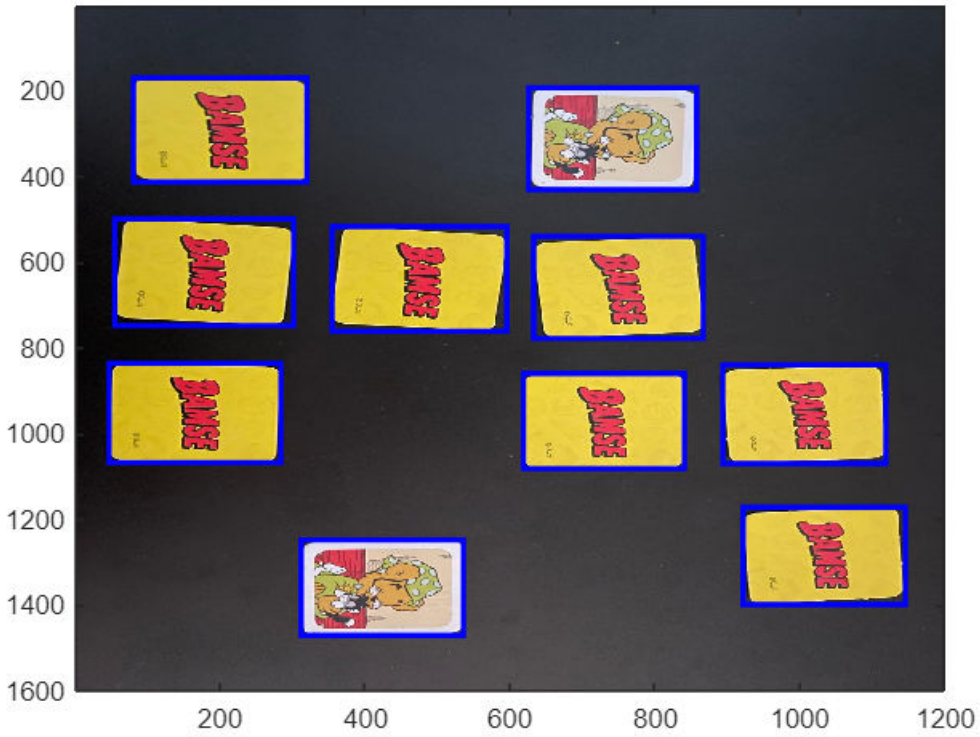


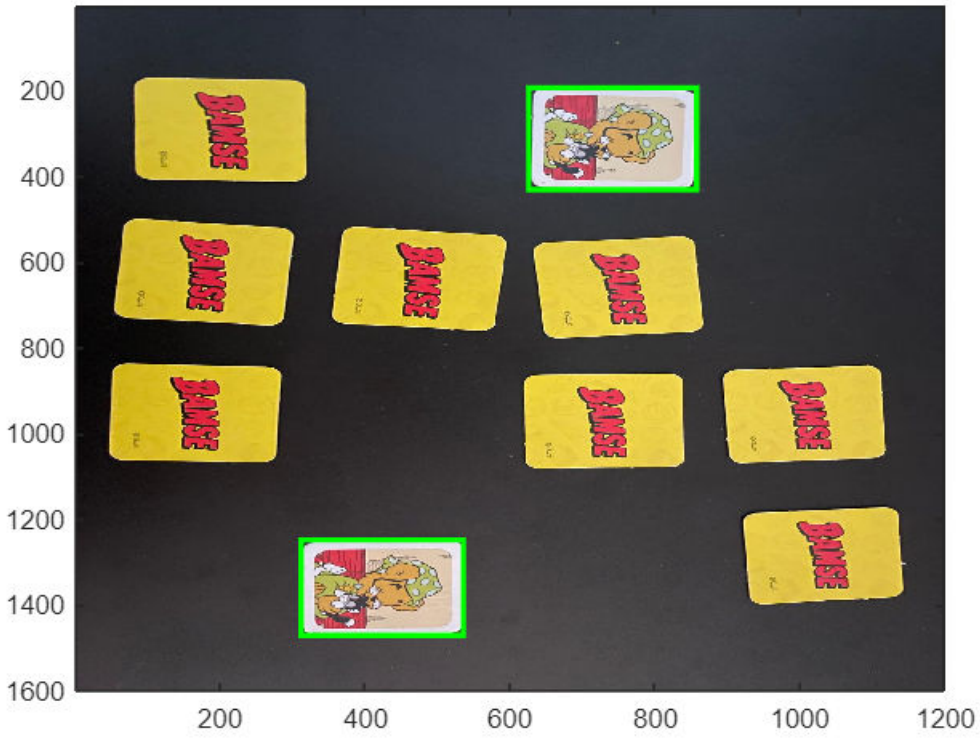
No match! Turn the two cards face down again. Player 1 may then continue



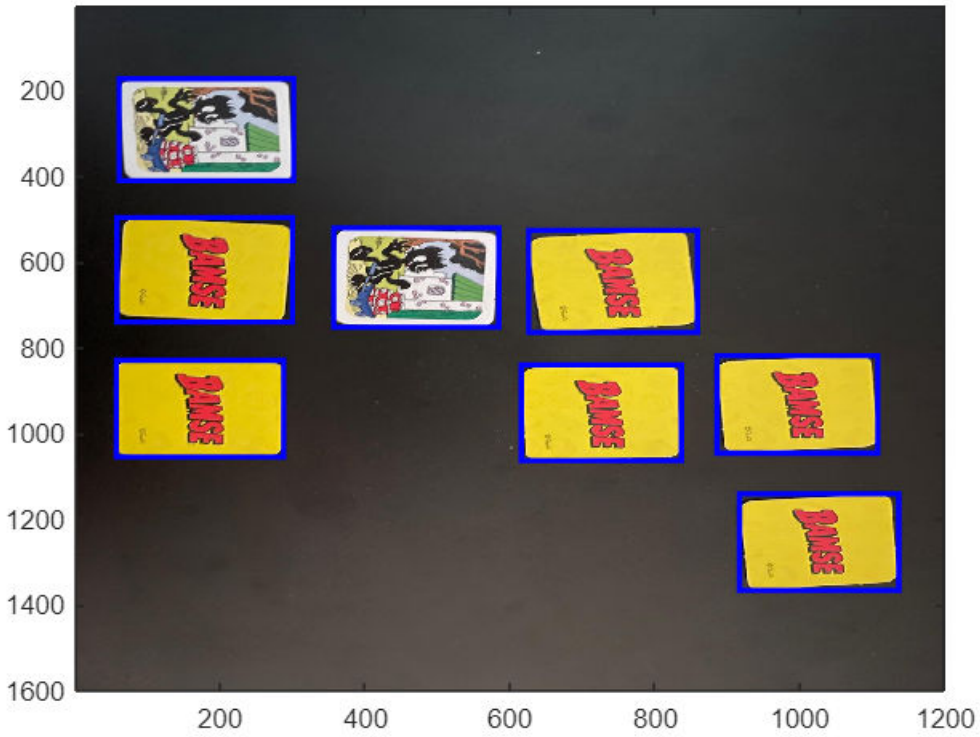


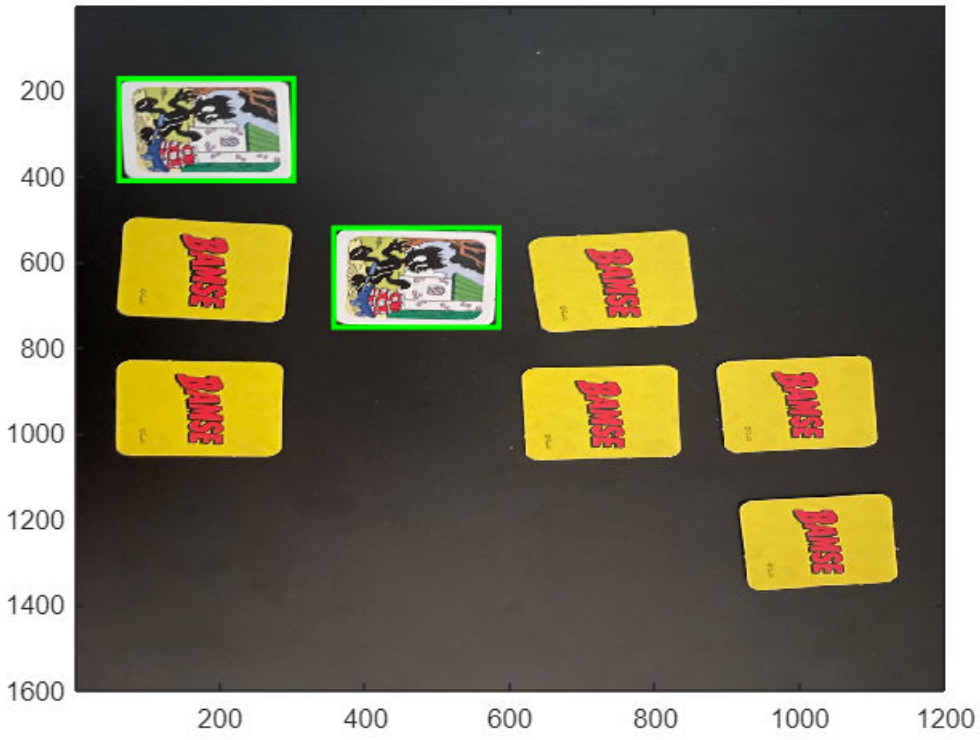
No match! Turn the two cards face down again. Player 2 may then continue



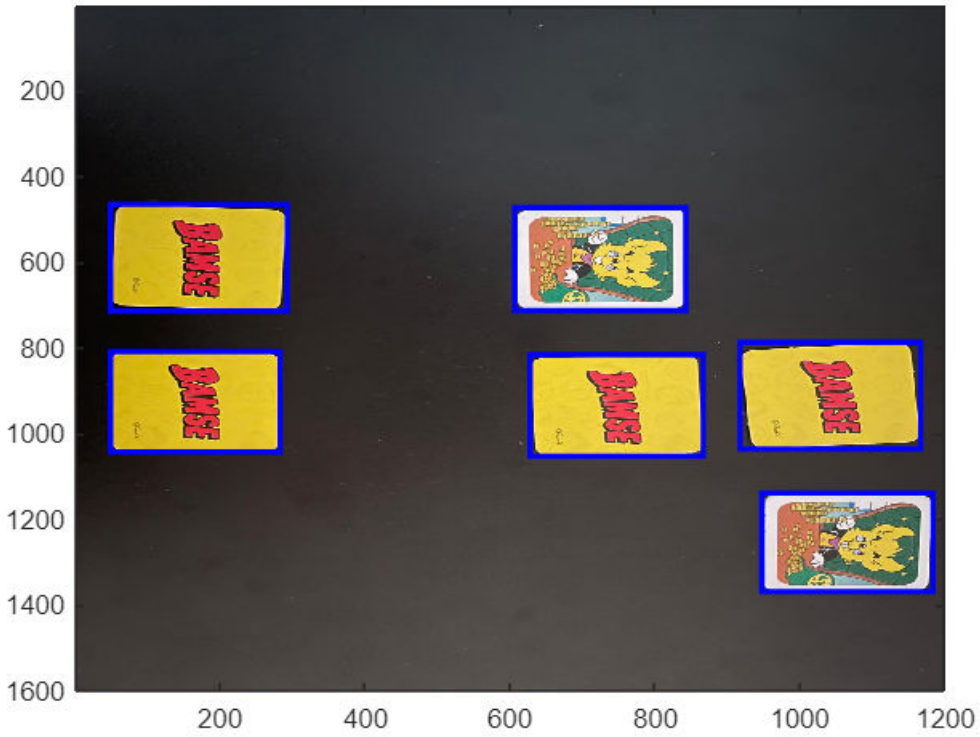


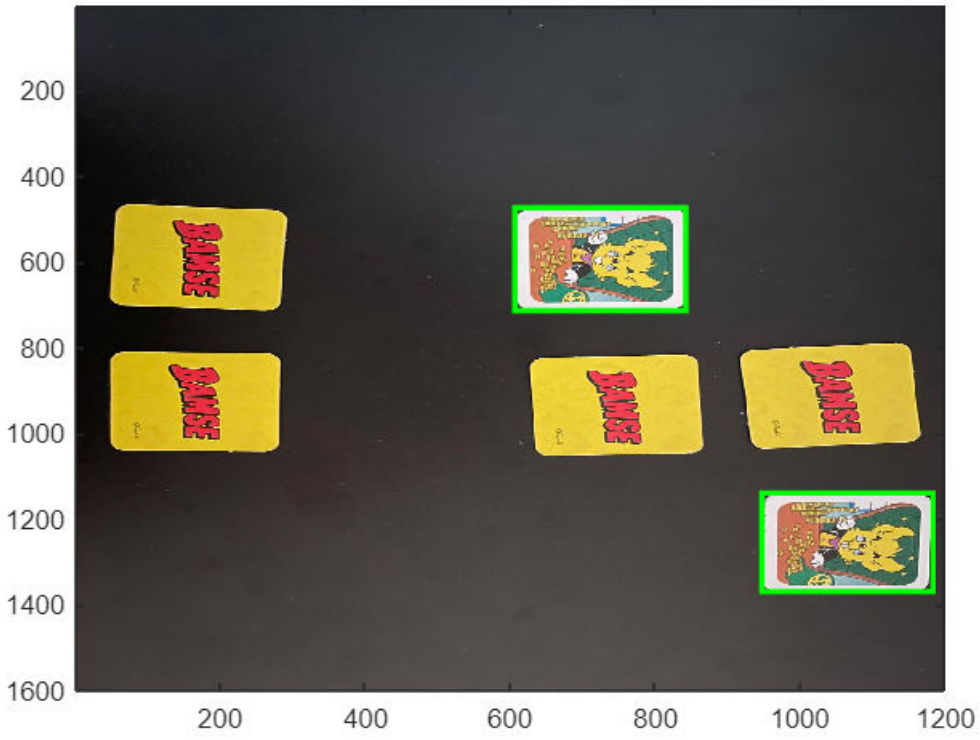
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



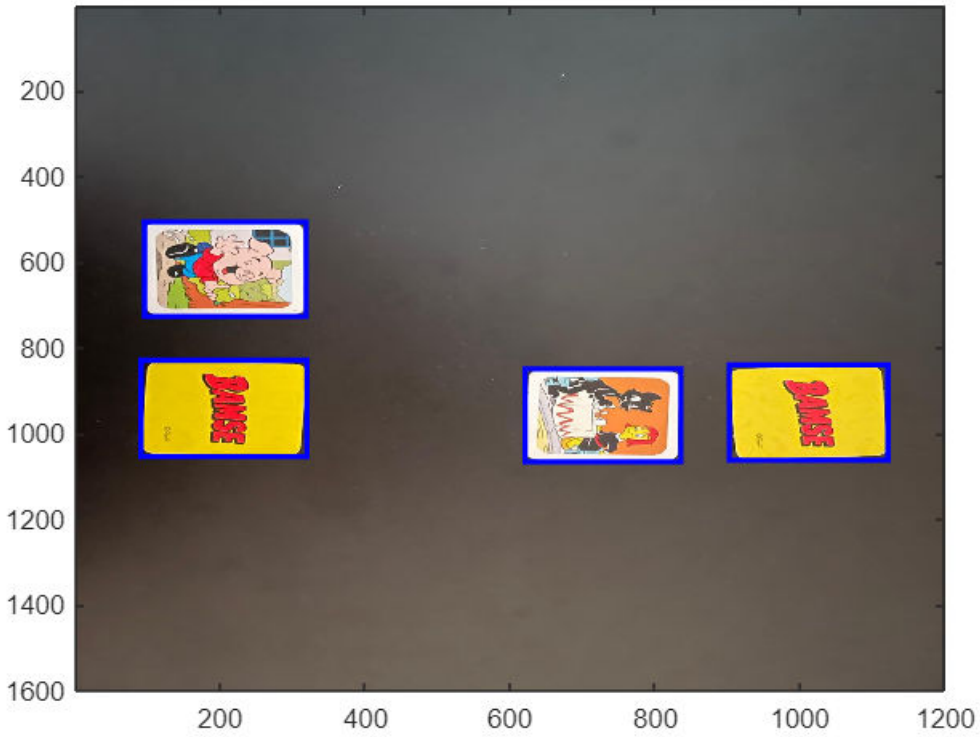


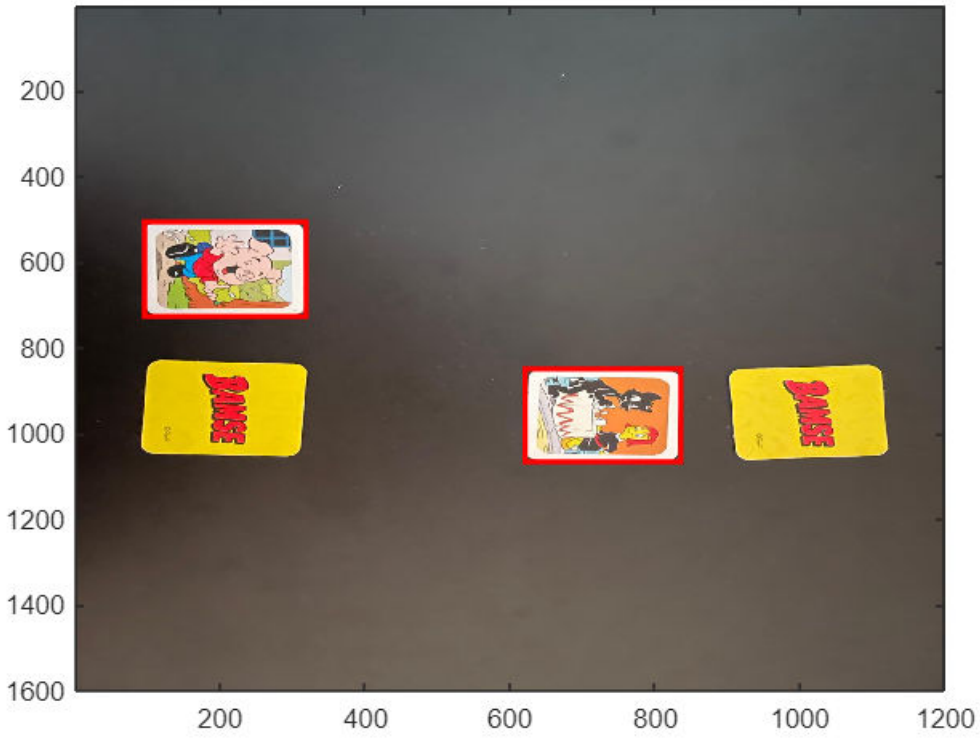
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



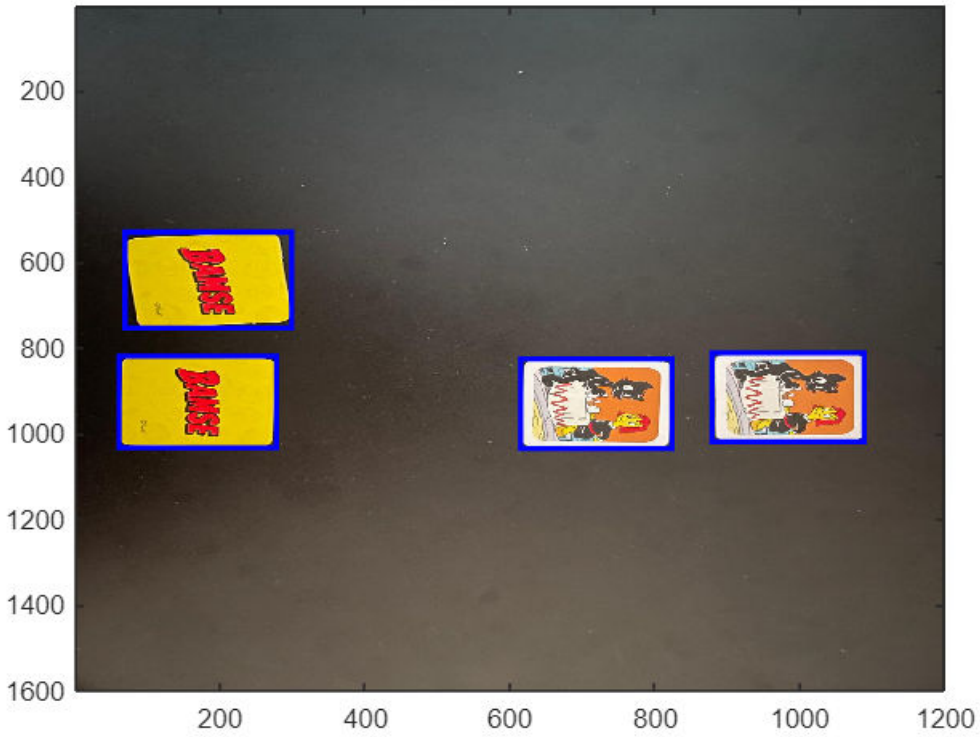


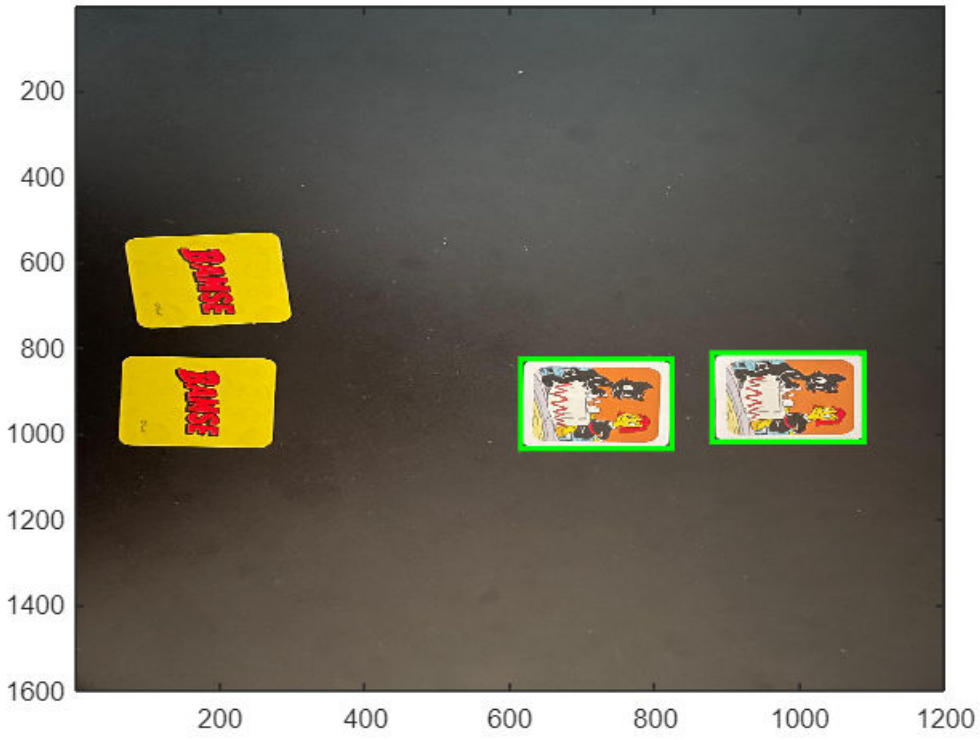
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



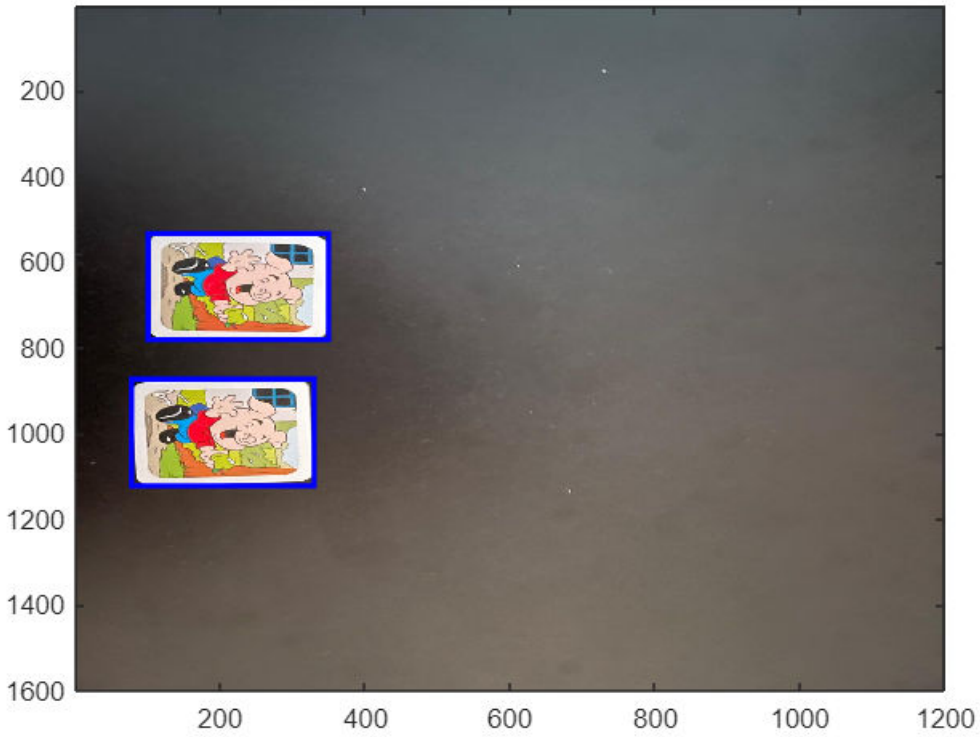


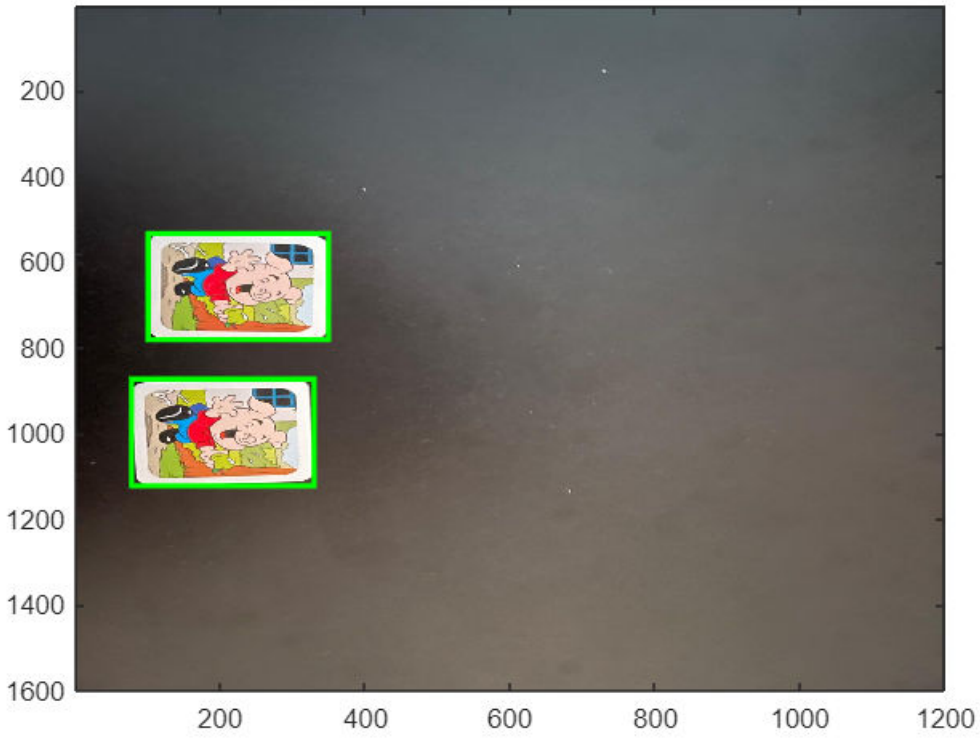

```
No match! Turn the two cards face down again. Player 1 may then continue
```





Congratulations! A matching pair. Please pick up your matching card pair and then you may continue





Congratulations! A matching pair. Please pick up your matching card pair and then you may continue

Declaring a winner

Once all of the cards have been turned and a winner has been found, the winner is declared. If both players have the same amount of matches, it is a draw.

```
declare_winner(score_1,score_2);
```

```
It is a draw, both players have 4 points.
```

Advanced part

```
clear all
```

Loading the images

The first step is to load the images again, this time images were taken of a cards that were scattered randomly. In addition, the images were taken from an angle and not directly above the cards.

```
images = load_images_advanced();
```

Game set up

Just like with the mandatory part, first the cards are shown all laying face down and boundaries are drawn around the cards.

```
img_n = 1;              % Start with the first image
image = images{img_n};
im = rgb2gray(image);

% Turn the image into a binary image
binary_im = binarize_image(image);

% Create regions around the cards
stats = regionprops(binary_im,'Boundingbox','Area','Centroid', 'Orientation');

% Draw boundaries around the cards in the first image
boundaries_first_image(binary_im,image);
```


```
Player 1, please choose two cards to turn
```

Playing the game

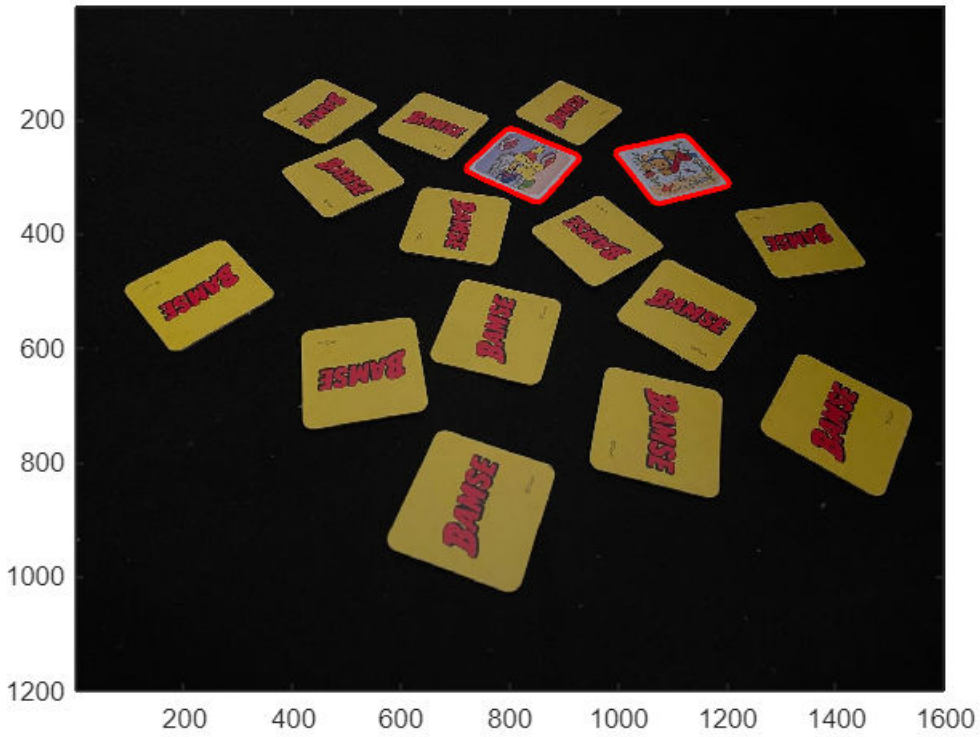
Now the two players turn two cards at the time, if they find a match, they may continue, otherwise the other player can play.

```
% Initialize the scores and the players
current_player = 1;
score_1 = 0;
score_2 = 0;

for n = 2:length(images)
    % Initialize the variables
    box = [];
    card_up = [];
    card_down = [];
    matches_up = [];
    box_c = [];

    % Take the current image
```

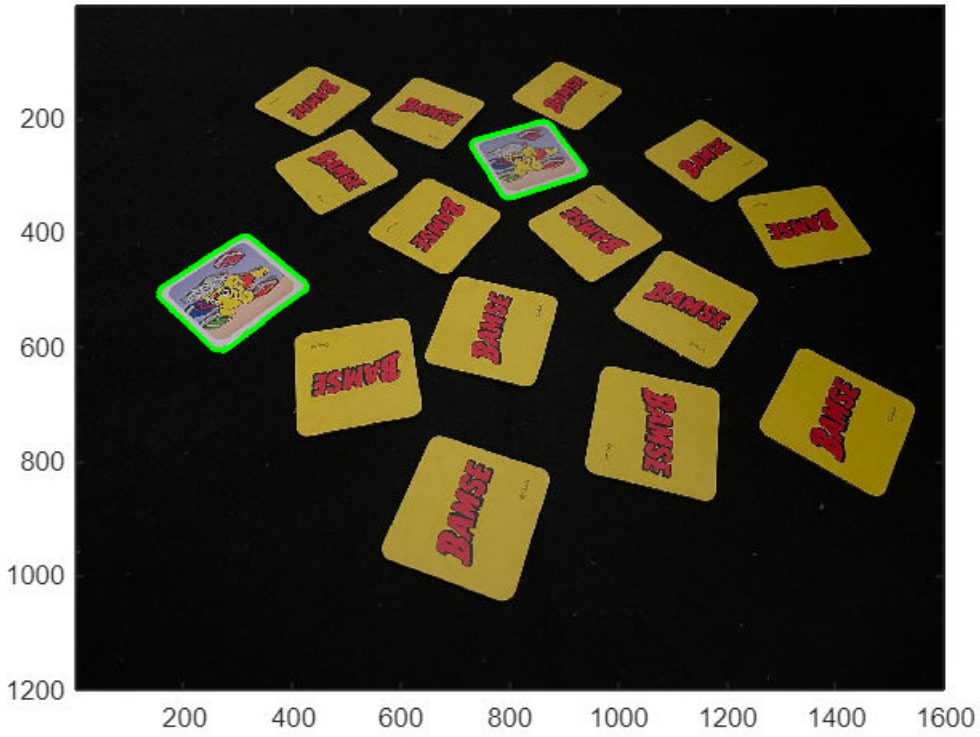
```
    image = images{n};

    % Create a binary image of the image
    binary_im = binarize_image(image);

    % create bounding boxes around cards
    stats_c = regionprops(binary_im,'Boundingbox','Area','Centroid', 'Orientation');

    % Take only bounding boxes around the cards it self by checking that
    % area is large enough to be around a card
    c = 1;
    for i = 1:length(stats_c)
        if stats_c(i).Area > 10000
            box_c{c} = stats_c(i);
            c = c + 1;
        end
    end

    % Create masks over the cards in the bounding box
    stats_mask = create_mask_card(box_c, binary_im);

    % See which cards are up and which are down
    [card_up,card_down] = card_up_or_down_advaned(stats_mask, image);

    % Check if the cards are a match by looking at the amount of matching
    % SIFT features
    match = matching_cards_advanced(card_up,image);


    % Create boundaries around the face-up cards, the color depends on
    % whether or not they are a matching pair
    create_boundary_advanced(card_up,image,binary_im,match)

    % If there is a match, increase the score of the current player
    if match == 1
        [score_1,score_2] = players_score(current_player,score_1,score_2);
    end

    % Determine whether the current player is switched or not
    current_player = player_turn(match, current_player);


end
```

No match! Turn the two cards face down again. Player 2 may then continue

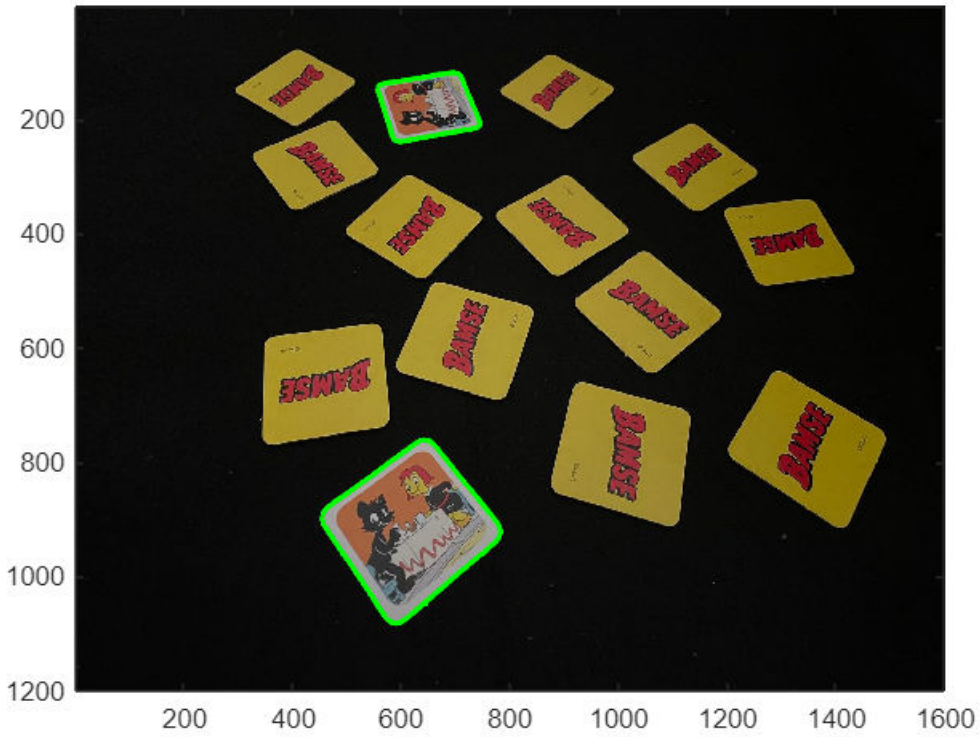
No match! Turn the two cards face down again. Player 1 may then continue



No match! Turn the two cards face down again. Player 2 may then continue

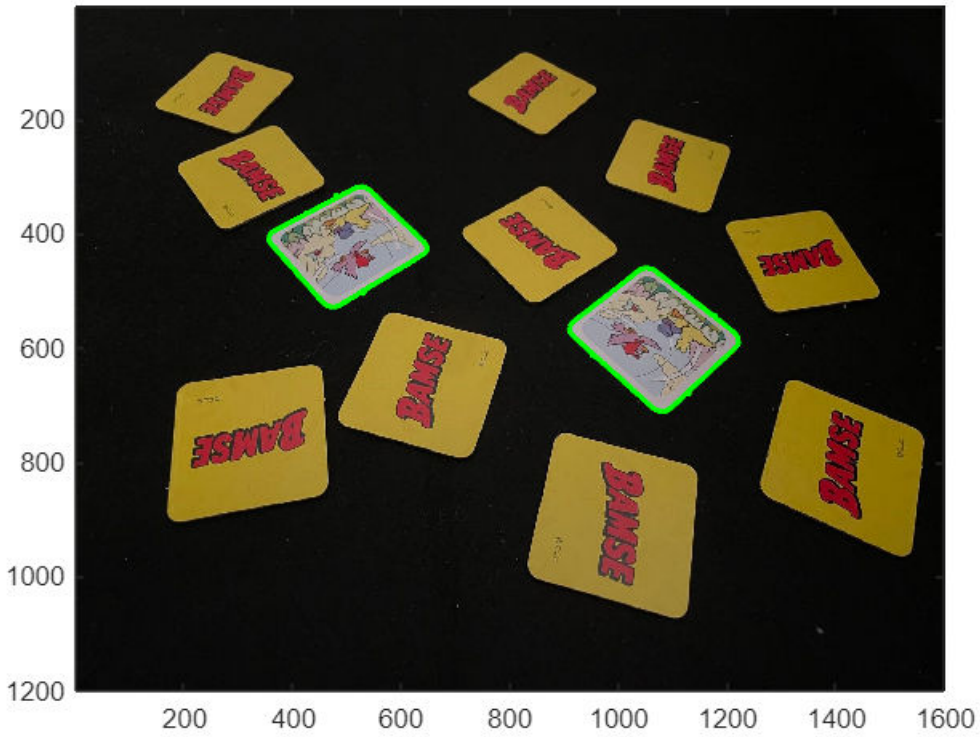


Congratulations! A matching pair. Please pick up your matching card pair and then you may continue

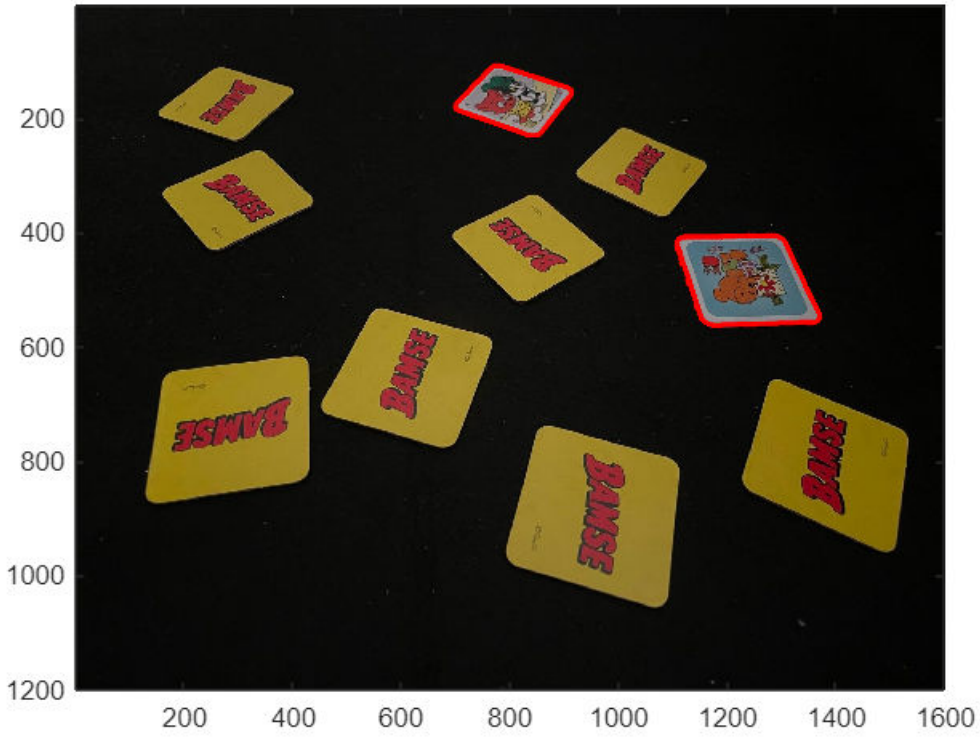


Congratulations! A matching pair. Please pick up your matching card pair and then you may continue

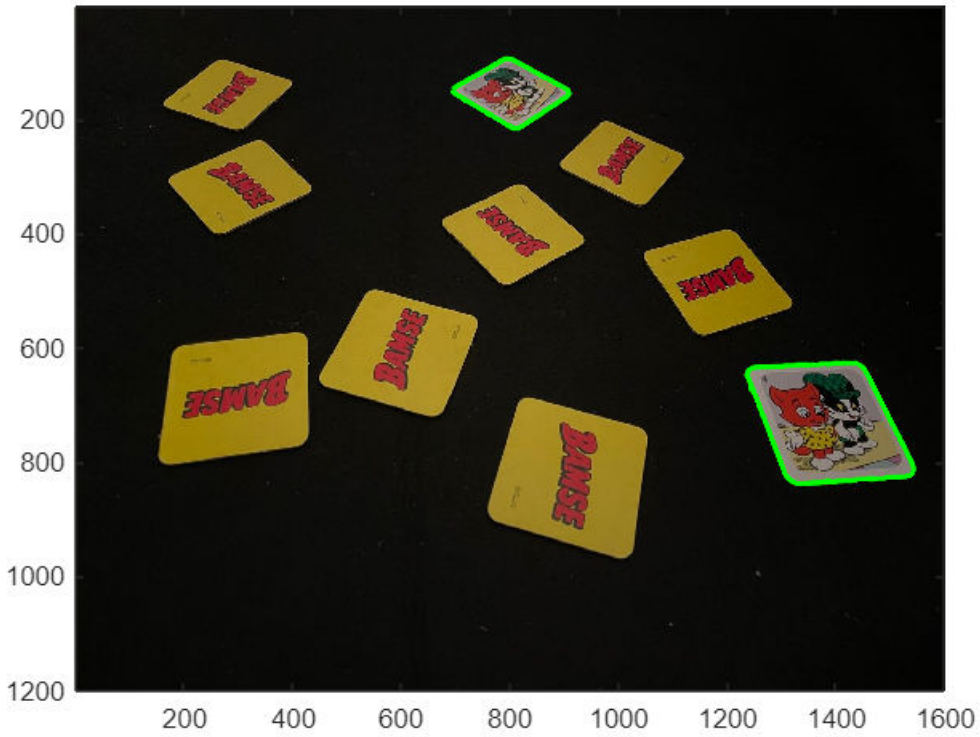
No match! Turn the two cards face down again. Player 1 may then continue



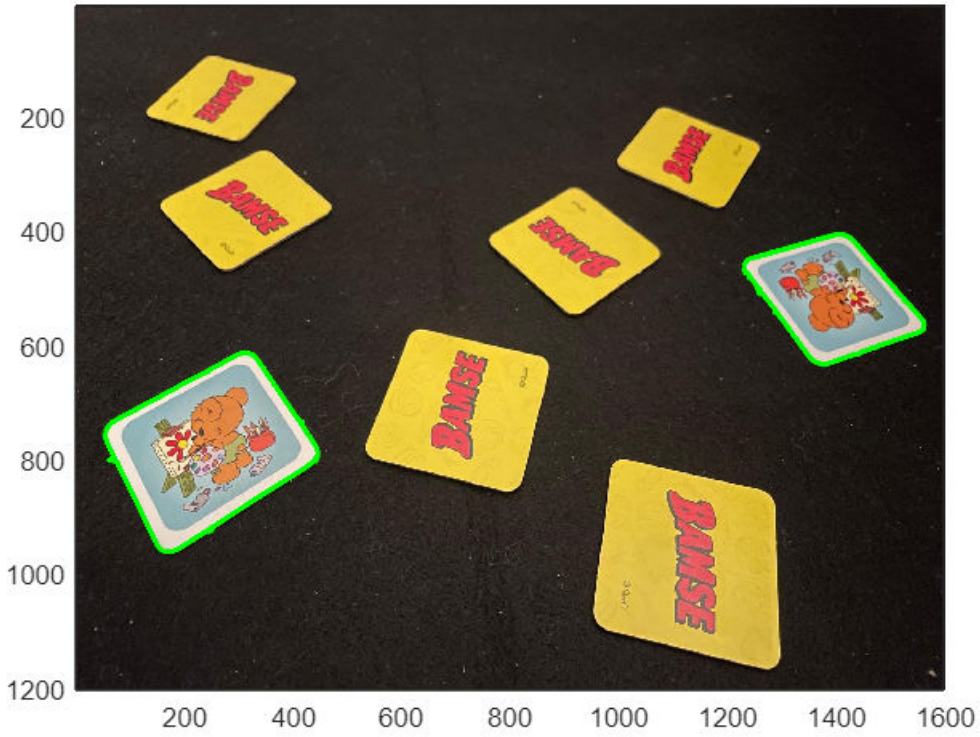
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



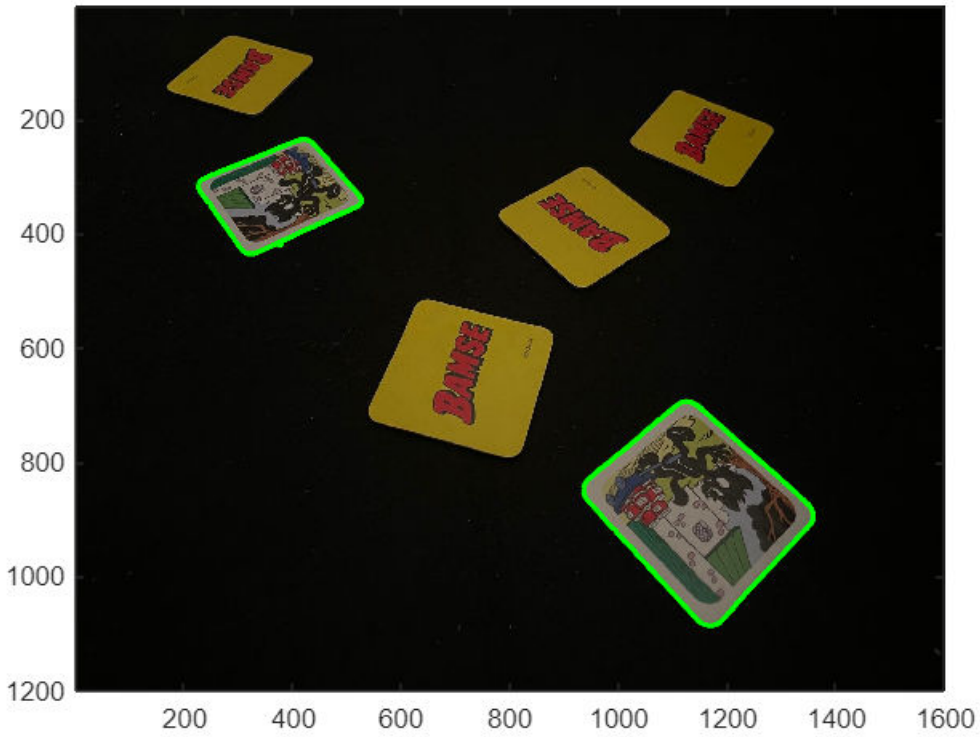
No match! Turn the two cards face down again. Player 2 may then continue



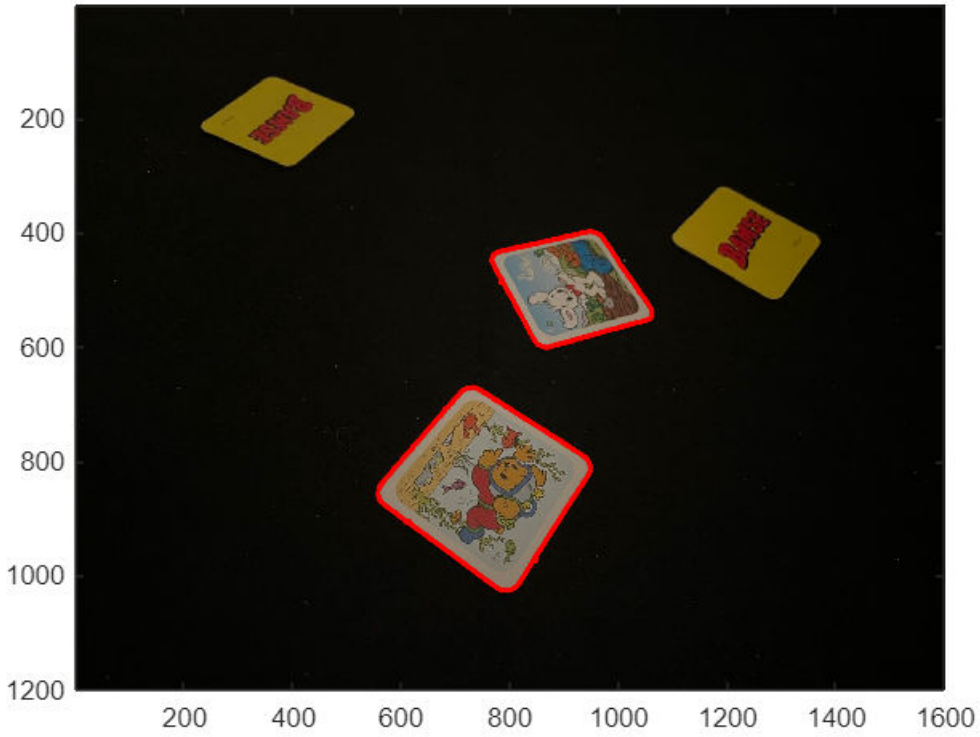
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



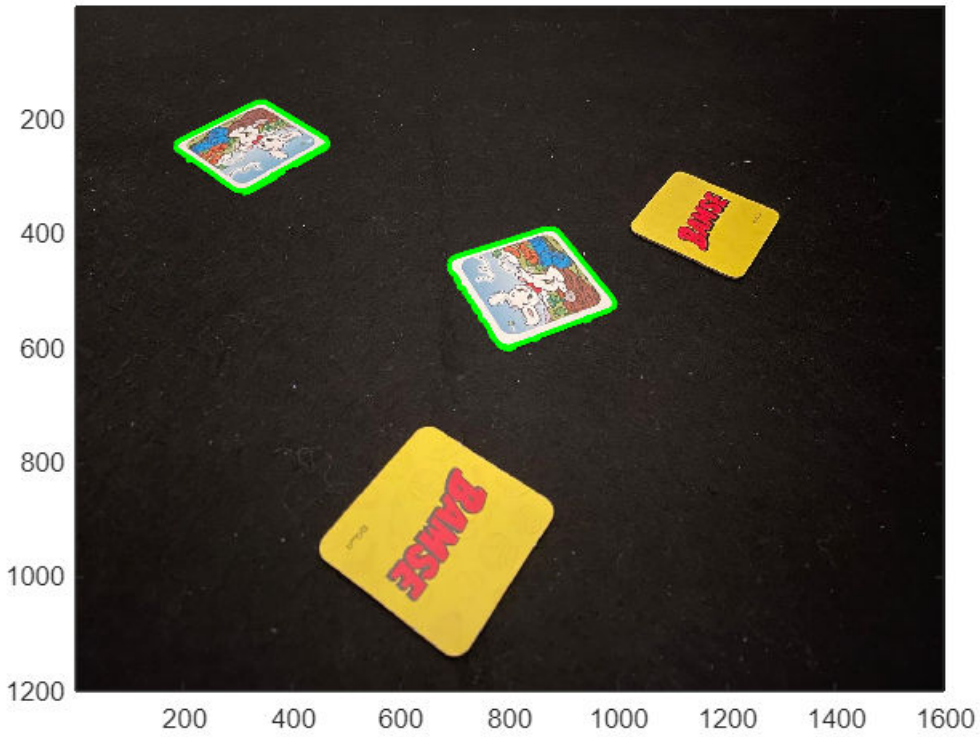
Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



Congratulations! A matching pair. Please pick up your matching card pair and then you may continue



No match! Turn the two cards face down again. Player 1 may then continue



Congratulations! A matching pair. Please pick up your matching card pair and then you may continue

Congratulations! A matching pair. Please pick up your matching card pair and then you may continue

Declearing a winner

After all of the matching pairs have been found, a winner is announced.

```
declare_winner(score_1,score_2);
```

```
Player 2 won the game with 5 points, congratulations!
```