Development Contest

Which Princess are You?

COMP4687.1
Introduction to Computer Vision

Once Upon a Time..

...there was a little girl who wondered which princess she was.

Snow White

Cinderella

Aurora

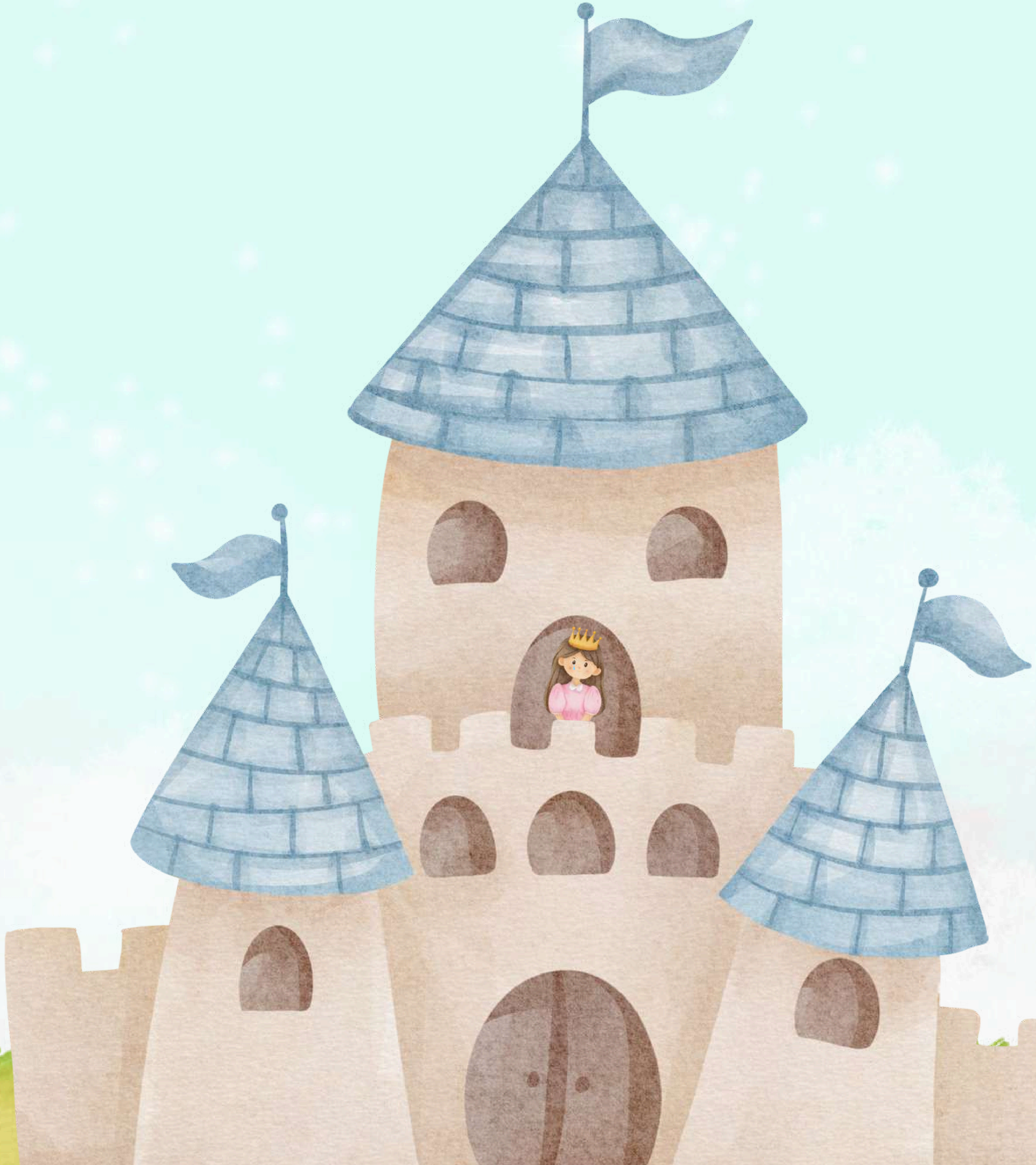
ARIEL

Belle

Jasmine

DISNEY
PRINCESS

Sadly, no more princesses. It’s time for the grownup responsibilities. We are going to review the codes now...



OpenCV Methods Used

1. detectMultiScale: Used for face detection.

2. calcHist: Calculates the image histogram.

3. compareHist: Compares histograms.

4. cvtColor: Changes the image color format.

5. resize: Equalizes the image dimensions.

```
static {
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
}
```

- Loads OpenCV's native libraries. This is required to use OpenCV functions in Java.

```
Mat userImage = Imgcodecs.imread(userPhotoPath);
if (userImage.empty()) {
System.out.println("User photo could not be loaded!");
return null;
}
```

```
MatOfRect userFaces = new MatOfRect();
faceDetector.detectMultiScale(userImage, userFaces);
```

- Reads the user's photo and detects the face using the Haar Cascade model.

```
Rect userFace = userFaces.toArray()[0];
Mat userFaceImage = new Mat(userImage, userFace);
```

- Rect: Holds the coordinates of the face region.
- Mat: Creates a new image from the face region.
- Takes the rectangle of the first detected face and crops this region.

```
File[] princessFiles = folder.listFiles((dir, name) ->
name.endsWith(".png"));
```

- Loads princess images with .png extension from the specified folder.

```
Imgproc.resize(princessImage, resizedPrincessImage,
userFaceImage.size());
```

- Equalizes the size of princess images with the user's face photo.

```
double similarity =
compareHistograms(userFaceImage,
resizedPrincessImage);
```

- Calculates the similarity score by comparing the histograms of user face and princess images.

```
if (similarity > bestMatchScore) {
bestMatchScore = similarity;
bestMatchPrincess = princessFile.getName();
}
```

- Takes the highest score from the histogram comparison and determines the most similar princess accordingly.

```
public static void main(String[] args) {
PrincessMatcher matcher = new
PrincessMatcher();

String userPhotoPath = "src/user/user.png";
String imagePath = "src/images";
String haarCascadePath =
"src/resources/haarcascade_frontalface_default.
xml";

String bestMatchPrincess =
matcher.findMostSimilarPrincess(userPhotoPath,
imagePath, haarCascadePath);

if (bestMatchPrincess != null) {
System.out.println("Most similar princess: " +
bestMatchPrincess);
} else {
System.out.println("No match found or an error
occurred.");
}
}
```

- Executes the main function of the class. Defines the paths of the user and princess images and prints the match result.

The End

[redacted]