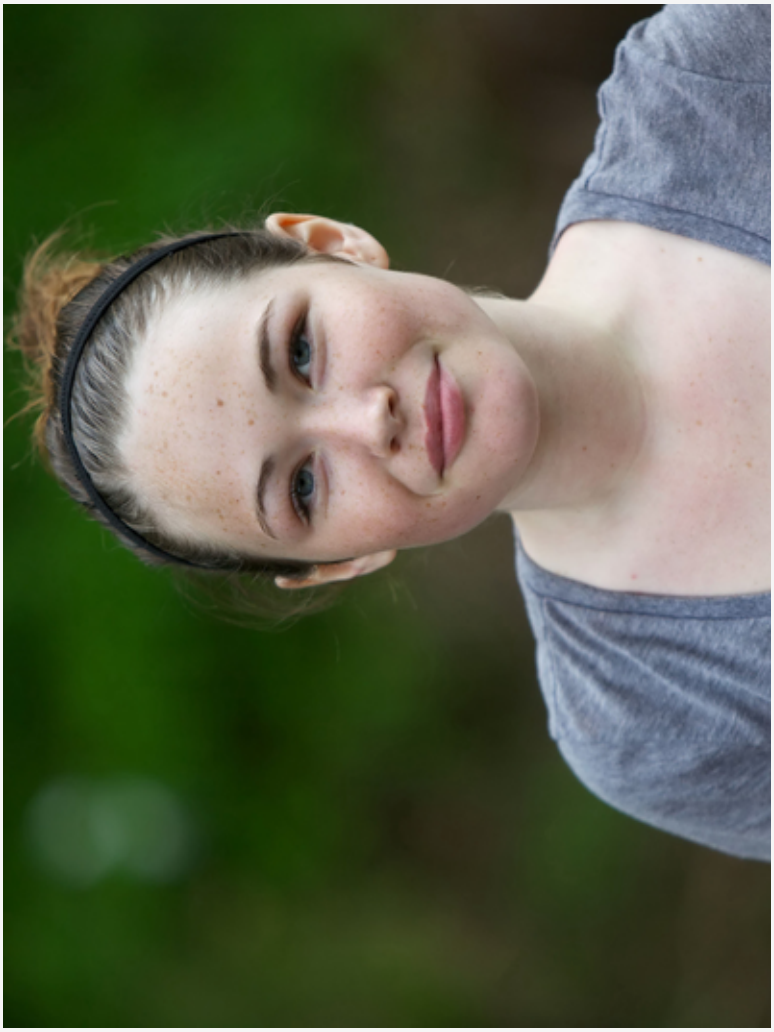
In[137]:=
```
(* Import image *)
imgOrig = Import["https://raw.githubusercontent.com/rekby/examples/
      e144648a8d5b065472bf3501f0551360d4ec6658/wolfram-mathematica/images/
      rotated-face.jpg"];
```

In[138]:=
```
(* Resize for better show *)
img = ImageResize[imgOrig, 400];

(* Check about can't find face (version 13.0.1) *)
HighlightImage[img, FindFaces]
```

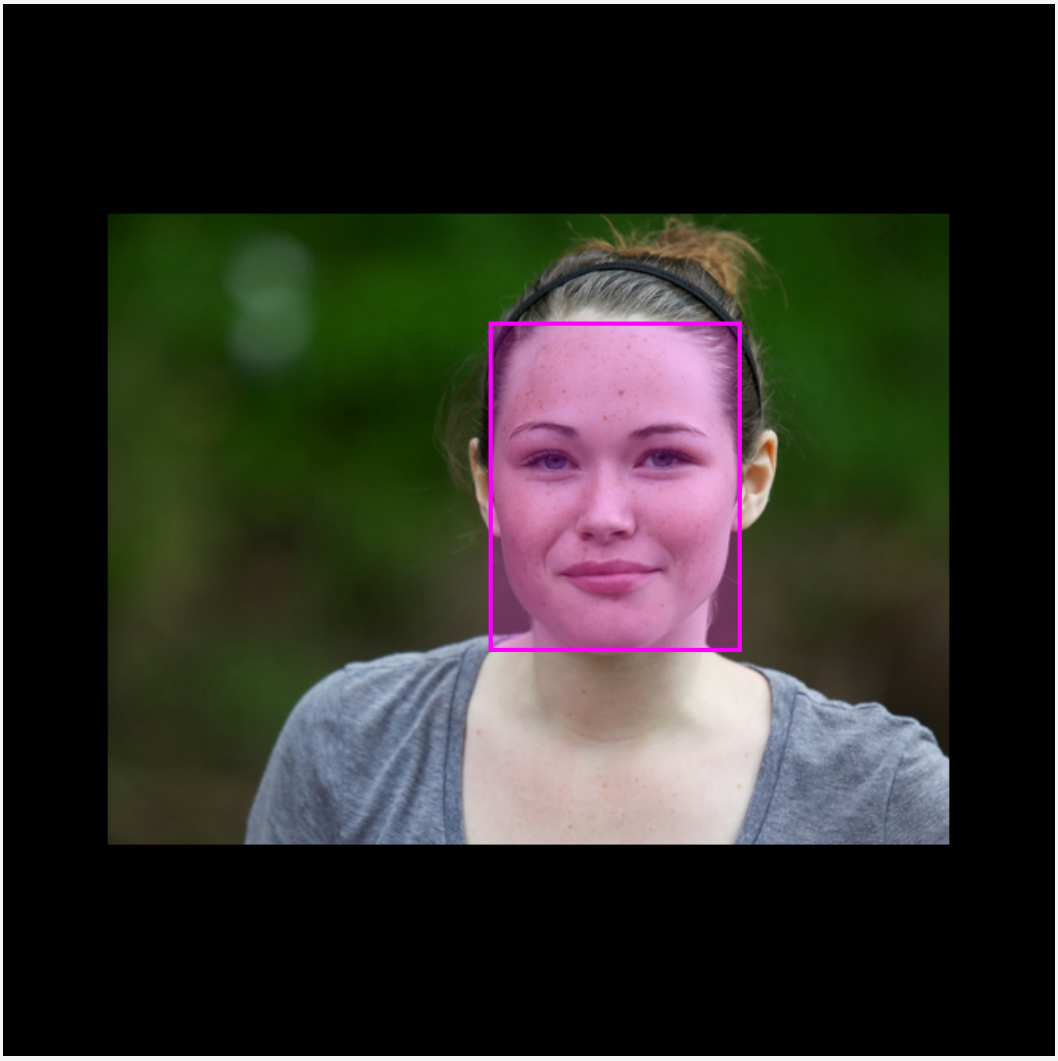
Out[139]=



In[140]:=
```
(* Rotate image for find face.
Size All need for got square image with same center for any rotation angle
it allow simple correct coordinats while re-
 calculate coorinations from rotated
to origin image.
If skip All argument - image will crop after rotate and it
  will difficult right reverse rotate recognized rectangle. For
  square images it is simple - rotate around center image.
*)
rotatedImage = ImageRotate[img, -Pi / 2, All];
rotatedFaces = FindFaces[rotatedImage];

(* Check about face found in rotated image*)
HighlightImage[rotatedImage, rotatedFaces]
```

Out[142]=



In[143]:=
```
(* remember face rectangle *)
rotatedFace = rotatedFaces[[1]];
```

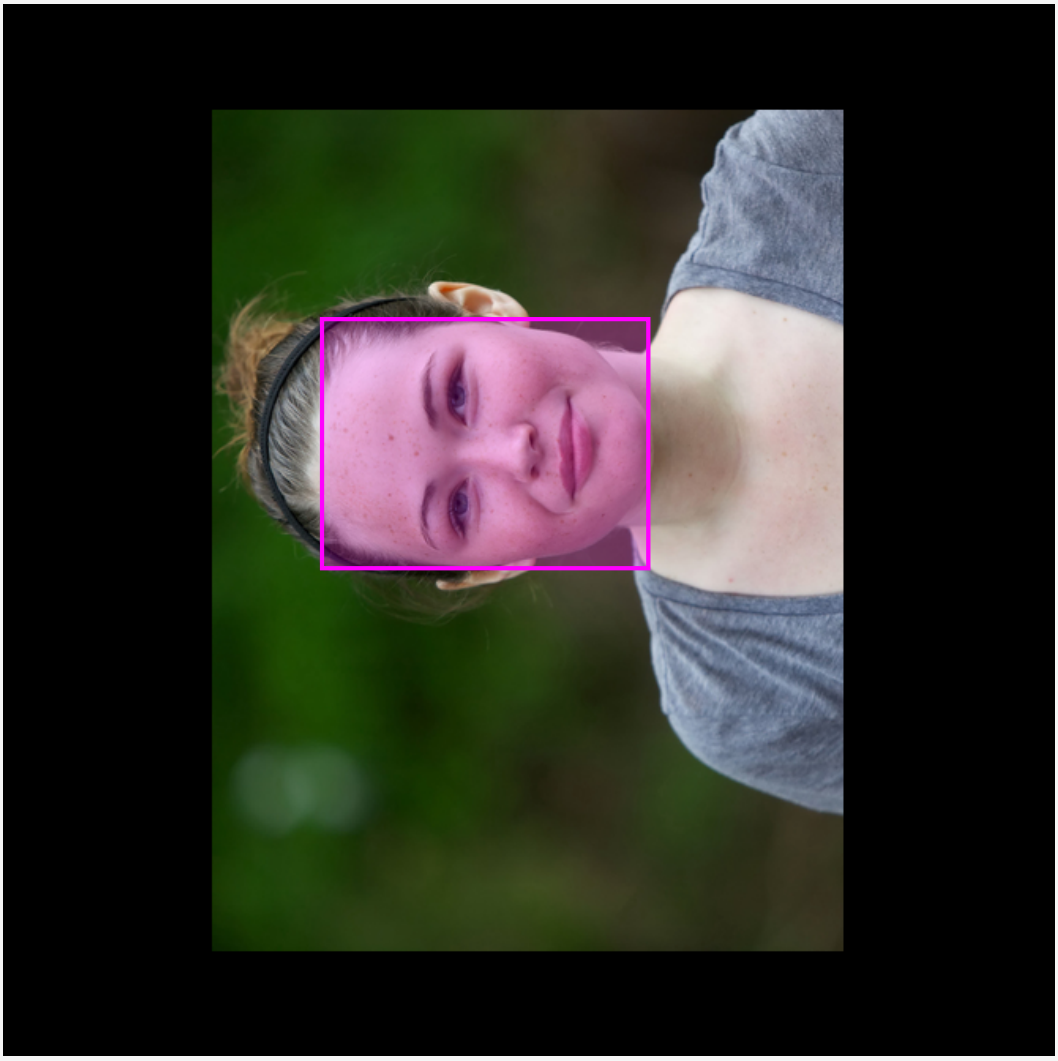
In[144]:=
```
(* Convert to original image coordinates.
If image rotated to 90° (as in example) reverse rotate may done with simple
 coordination arifmetic. Next code will work for any rotation angle.
*)

(* rotate face rectangle around center for
 place in right place in resized origin photo *)
rotatedSize = ImageDimensions[rotatedImage];
```

In[145]:=
```
center = rotatedSize / 2;
```

In[146]:=
```
rt = RotationTransform[Pi / 2, center];
originFace = rt[rotatedFace];
HighlightImage[ImageRotate[img, 0, All], originFace]
```

Out[148]=



In[149]:=
```
(* Then need fix coordinates for original not rotate-resized image *)
originFaceCoordinates = PolygonCoordinates[originFace];
 imgSize = ImageDimensions[img];

correctToOrigin = (rotatedSize - imgSize) / 2;
```

In[152]:=
```
movedFace = (#1 - correctToOrigin) & /@ originFaceCoordinates;

(* Use BoundingRegion because oordinates from poligon got in bad order
 and Poligon function create bad form polygon instead of rectangle *)
movedFacePoligon = BoundingRegion[movedFace];

(* Visual check about coordinates correct *)
HighlightImage[img, {movedFacePoligon}]
```

Out[154]=