Reproducible Methods for Face Research

Lisa DeBruine[1], Iris Holzleitner[1,2], & Benedict C. Jones[3]

[1] Institute of Neuroscience & Psychology, University of Glasgow
[2] University of the West of England, Bristol
[3] University of Strathclyde, Glasgow

Draft

Abstract

Face stimuli are commonly created in non-reproducible ways. In this paper, we document the irreproducibility of much existing work on faces, explain the benefits of reproducible stimuli, and introduce the open-source R package webmorphR. We explain the technical processes of morphing and transforming through a case study of creating face stimuli from an open-access image set. Finally, we discuss some ethical and methodological issues around the use of face images in research that may be ameliorated through the use of reproducible stimuli.

Keywords: faces; morphing; transforming; reproducibility; webmorph
Word count: X

Introduction

Face stimuli are commonly used in research on visual and social perception. This almost always involves some level of stimulus preparation to rotate, resize, crop, and reposition faces on the image. In addition, many studies systematically manipulate face images by changing color and/or shape properties (e.g., A. L. Jones & Jaeger, 2019; reviewed in Little et al., 2011).

This research was funded by ERC grant #647910 (KINSHIP).

The authors made the following contributions. Lisa DeBruine: Conceptualization, Funding acquisition, Methodology, Software, Validation, Visualization, Writing - Original Draft Preparation; Iris Holzleitner: Software, Validation, Visualization, Writing - Review & Editing; Benedict C. Jones: Conceptualization, Writing - Review & Editing.

Correspondence concerning this article should be addressed to Lisa DeBruine, 62 Hillhead Street, Glasgow G12 8QB, Scotland. E-mail: lisa.debruine@glasgow.ac.uk

Over a decade ago, Gronenschild and colleagues (2009) argued for the importance of standardizing face stimuli so that they are not "confounded by factors such as brightness and contrast, head size, hair cut and color, skin color, and the presence of glasses and earrings". They describe a three-step standardization process. First, they manually removed features such as glasses and earrings in Photoshop. Second, they geometrically standardized images by semi-automatically defining eye and mouth coordinates used to fit the images within an oval mask, Third, they optically standardized images by converting them to greyscale and remapping values between the minimum and 98% threshold onto the full range of values. While laudable in its aims, this procedure has not achieved widespread adoption, probably because the authors provided no code or tools. In personal communication, the main author said that this is because "the procedure is based on standard image processing algorithms described in many textbooks". However, we were unable to easily replicate the procedure and found several places where instructions had more than one possible interpretation or relied on the starting images having specific properties, such as symmetric lighting reflections in the eyes. Additionally, greyscale images with an oval mask are not appropriate for many research questions.

The goal of this paper is to argue for the importance of reproducible stimulus processing methods in face research and to introduce an open-source R package that allows researchers to create face stimuli with scripts that can then be shared so that others can create stimuli using identical methods.

Why are reproducible stimulus construction methods important? Lisa once gave up on a research project because she couldn't figure out how to manipulate spatial frequency to make the stimuli look like those in a relevant paper. When she contacted the author, they didn't know how the stimuli were created because a postdoc had done it in Photoshop and didn't leave a detailed record of the method.

Reproducibility is especially important for face stimuli because faces are sampled, so replications should sample new *faces* as well as new participants (Barr, 2007). The difficulty of creating equivalent face stimuli is a major barrier to this, resulting in stimulus sets that are used across dozens or hundreds of papers. For example, the Chicago Face Database (Ma et al., 2015) has been cited in almost 800 papers. Ekman and Friesen's (1976) Pictures of Facial Affect has been cited more than 5500 times. This image set is currently **selling** for $399 for "110 photographs of facial expressions that have been widely used in cross-cultural studies, and more recently, in neuropsychological research". Such extensive reuse of image sets means that any confounds present in the image set can cause highly "replicable" but potentially false findings.

Additionally, image sets are often private and reused without clear attribution. Our group has only recently been trying to combat this by making image sets public and citable where possible (L. DeBruine, 2016; L. M. DeBruine & Jones, 2017a; e.g., L. M. DeBruine & Jones, 2017b; L. DeBruine & Jones, 2020; Morrison et al., 2018) and including clear explanations of reuse where not possible Holzleitner et al. (2019).

Common Techniques. In this section, we will give an overview of common techniques used to process face stimuli across a wide range of research involving faces. It was basically impossible to systematically survey the literature about the methods used to create facial stimuli, in large part because of poor documentation. However, several common methods are discussed below.

Mystery Methods. Many researchers describe image manipulation generically or use "in-house" methods that are not well specified enough for another researcher to have any chance of replicating them.

> Each of the images was rendered in gray-scale and morphed to a common shape using an in-house program based on bi-linear interpolation (see e.g., Gonzalez & Woods, 2002). Key points in the morphing grid were set manually, using a graphics program to align a standard grid to a set of facial points (eye corners, face outline, etc.). Images were then subject to automatic histogram equalization. (Burton et al., 2005, p. 263)

The reference above (Gonzalez et al., 2002) is a 190-page textbook. It mentions bilinear interpolation on pages 64–66 in the context of calculating pixel color when resizing images and it's unclear how this could be used to morph shape.

> They were cropped such that the hair did not extend well below the chin, resized to a height of 400 pixels, and placed on 400 x 400 pixel backgrounds consisting of phase-scrambled variations of a single scene image (for example stimuli, see Figure 1). (Pegors et al., 2015, p. 665)

While the example images in the figure mentioned above help to clarify the methods, it was clear that there was a large degree of subjectivity in determining how to crop the hair.

Photoshop/Image editors. A search for "Photoshop face attractiveness" produced 19,300 responses in Google Scholar[1]. Here are descriptions of the use of Photoshop from a few of the top hits.

> If necessary, scanned pictures were rotated slightly, using Adobe Photoshop software, clockwise to counterclockwise until both pupil centres were on the same y-coordinate. Each picture was slightly lightened a constant amount by Adobe Photoshop. (Scheib et al., 1999, p. 1914)

> These pictures were edited using Adobe Photoshop 6.0 to remove external features (hair, ears) and create a uniform grey background. (Sforza et al., 2010, p. 150)

> The averaged composites and blends were sharpened in Adobe Photoshop to reduce any blurring introduced by blending. (Rhodes et al., 2001, p. 615)

[1] All web search figures are from Google Scholar in May 2022.

Most papers that use Photoshop methods simply state in lay terms what the editing accomplished, and not the specific tools or methods in the application used to accomplish it. For example, it is not clear what sharpening tool was used in the last quote above, and what settings were used. Were all images sharpened by the same amount or was this done "by eye"?

A potential danger to processing images "by eye" is the possibility of visual adaptation affecting the researcher's perception. It is well known that viewing images with specific alterations to shape or colour alters the perception of subsequent images (Rhodes, 2017). Thus, a researcher's perception of the "typical" face can change after exposure to altered faces (L. isa M. DeBruine et al., 2007). While some processing will always require human intervention, reproducible methods can also allow researchers to record their specific decisions so such biases can be detected and corrected for.

Scriptable Methods. There are several scriptable methods for creating image stimuli, including MatLab, ImageMagick, and GraphicConvertor. Photoshop is technically scriptable, but a search of "Photoshop script face" only revealed a few computer vision papers on detecting photoshopped images (e.g., Wang et al., 2019).

MatLab (Higham & Higham, 2016) is widely used within visual psychophysics. A Google Scholar search for "MatLab face attractiveness" returned 23,000 hits, although the majority of papers we inspected used MatLab to process EEG data, present the experiment, or analyse image color, rather than using MatLab to create the stimuli. "MatLab face perception" generated 97,300 hits, more of which used MatLab to create stimuli.

> The average pixel intensity of each image (ranging from 0 to 255) was set to 128 with a standard deviation of 40 using the SHINE toolbox (function lumMatch) (Willenbockel et al., 2010) in MATLAB (version 8.1.0.604, R2013a). (Visconti di Oleggio Castello et al., 2014, p. 2)

ImageMagick (The ImageMagick Development Team, 2021) is a free, open-source program that creates, edits, and converts images in a scriptable manner. The {magick} R package (Ooms, 2021) allows you to script image manipulations in R using ImageMagick.

> Images were cropped, resized to 150 × 150 pixels, and then grayscaled using ImageMagick (version 6.8.7-7 Q16, x86_64, 2013-11-27) on Mac OS X 10.9.2. (Visconti di Oleggio Castello et al., 2014, p. 2)

GraphicConvertor (Nishimura, 2000) is typically used to batch process images, such as making images a standard size or adjusting color. While not technically "scriptable", batch processing can be set up in the GUI interface and then saved to a reloadable ".gaction" file. (A search for ' "gaction" GraphicConvertor' on Google Scholar returned no hits.)

> We used the GraphicConverterTM application to crop the images around the cat face and make them all 1024x1024 pixels. One of the challenges of image matching is to do this process automatically. (Paluszek & Thomas, 2019, p. 214)

Scriptable methods are a laudable start to reproducible stimuli, but the scripts themselves are often not shared, or are in a proprietary closed format, such as MatLab. Additionally, most images that were processed with scriptable methods also used some non-scripted pre-processing to manually crop or align the images.

Commerical morphing. Face averaging or "morphing" is a common technique for making images that are blends of two or more faces. We found 937 Google Scholar responses for "Fantamorph face", 170 responses for "WinMorph face" and fewer mentions of several other programs, such as MorphThing (no longer available) and xmorph.

Most of these programs do not use open formats for storing delineations: the x- and y-coordinates of the landmark points that define shape and the way these are connected with lines. Their algorithms also tend to be closed and there is no common language for describing the procedures used to create stimuli in one program in a way that is easily translatable to another program. Here are descriptions of the use of commercial morphing programs from a few of the top hits.

> The faces were carefully marked with 112 nodes in FantaMorph™, 4th version: 28 nodes (face outline), 16 (nose), 5 (each ear), 20 (lips), 11 (each eye), and 8 (each eyebrow). To create the prototypes, I used FantaMorph Face Mixer, which averages node locations across faces. Prototypes are available online, in the Personality Faceaurus [http://www.nickholtzman.com/faceaurus.htm]. (Holtzman, 2011a, p. 650)

The link above contains only morphed face images and no further details about the morphing or stimulus preparation procedure.

> The 20 individual stimuli of each category were paired to make 10 morph continua, by morphing one endpoint exemplar into its paired exemplar (e.g. one face into its paired face, see Figure 1C) in steps of 5%. Morphing was realized within FantaMorph Software (Abrosoft) for faces and cars, Poser 6 for bodies (only between stimuli of the same gender with same clothing), and Google SketchUp for places. (Weigelt et al., 2013, p. 4)

Psychomorph/WebMorph. Psychomorph is a program developed by Benson, Perrett, Tiddeman and colleagues. It uses "template" files in a plain text open format to store delineations and the code is well documented in academic papers and available as an open-source Java package.

Benson and Perrett (Benson & Perrett, 1991a, 1991b, 1993) describe algorithms for creating composite images by marking corresponding coordinates on individual face images, remapping the images into the average shape, and combining the colour values of the remapped images. These images are also called "prototype" images and can be used to generate caricatures.

The averaging and caricaturing methods were later complemented by a transforming method (Rowland & Perrett, 1995). This method quantifies shape and colour differences between a pair of faces, creating a "face space" vector along which other faces can be manipulated. This method is distinct from averaging. For example, averaging an individual face with a prototype smiling face will produce a face that looks approximately halfway between the individual and the prototype. The smile will be more intense than the original individual's smile if they weren't smiling, and be less intense if the individual was smiling more than the prototype. However, the transform method defines the shape and/or color difference between neutral and smiling prototypes to define a vector of smiling. Transforming an individual face by some positive percent of the difference between neutral and smiling faces will then always result in an individual face that looks *more* cheerful than the original individual, no matter how cheerful they started out (Fig 1).

Figure 1. Composite (A) neutral and (B) smiling faces made from 49 indvidual neutral and smiling identities. (C) Individual smiling faces were (D) averaged with the smiling composite or (E) transformed by 50% of the shape and color differences between the neutral and smiling composites (E).

These methods were improved by wavelet-based texture averaging (Tiddeman et al., 2001), resulting in images with more realistic textural details, such as facial hair and eyebrows. This reduces the "fuzzy" look of composite images, but can also result in artifacts, such as lines on the forehead in Figure 2, which are a result of some images having a fringe.

The desktop version of Psychomorph was last updated in 2013, and can be difficult to install on some computers. To solve this problem, we started developing WebMorph (L. M. DeBruine, 2018), a web-based version that uses the Facemorph Java package from Psychomorph for averaging and transforming images, but has independent methods for delineation and batch processing. While the desktop version of Psychomorph has limited batch processing ability, it requires a knowledge of Java to be fully scriptable. WebMorph has more extensive batch processing capacity, including the ability to set up image processing scripts in a spreadsheet, but some processes such as delineation still require a fair amount of manual processing. In this paper, we introduce webmorphR (L. M. DeBruine, 2022a), an R package companion to WebMorph that allows you to create R scripts to fully and reproducibly describe all of the steps of image processing and easily apply them to a new set of images.

Figure 2. Untextured and textured prototypes of 4 male faces.

Table 1
Glossary of terms.

Term	Definition
composite	an average of more than one face image
delineation	the x- and y-coordinates for a specific template that describe an image
landmark	a point that marks corresponding locations on different images
lines	connections between landmarks; these may be used to interpolate new landmarks for morphi
morphing	blending two or more images to make an image with an average shape andor color
prototype	an average of faces with similar characteristics, such as expression, gender, age, and/or ethni
template	a set of landmark points that define shape and the way these are connected with lines; only i
transforming	changing the shape and/or color of an image by some proportion of a vector that is defined a

Methods

In this section, we will cover some common image manipulations and how to achieve them reproducibly using webmorphR (L. M. DeBruine, 2022a). We will also be using webmorphR.stim (L. DeBruine & Jones, 2017), a package that contains a number of open-source face image sets, and webmorphR.dlib (L. M. DeBruine, 2022b), a package that provides dlib models and functions for automatic face detection. These latter two packages cannot be made available on CRAN (the main repository for R packages) because of their large file size.

Editing. Almost all image sets start with raw images that need to be cropped, resized, rotated, padded, and/or color normalised. Although many reproducible methods exist to manipulate images in these ways, they are complicated when an image has an associated delineation, so webmorphR has functions that alter the image and delineation together (Fig. 3).

```
orig <- demo_stim() # load demo images
mirrored <- mirror(orig)
cropped  <- crop(orig, width = 0.75, height = 0.75)
resized  <- resize(orig, 0.75)
rotated  <- rotate(orig, degrees = 180)
padded   <- pad(orig, 30, fill = "black")
grey     <- greyscale(orig)
```

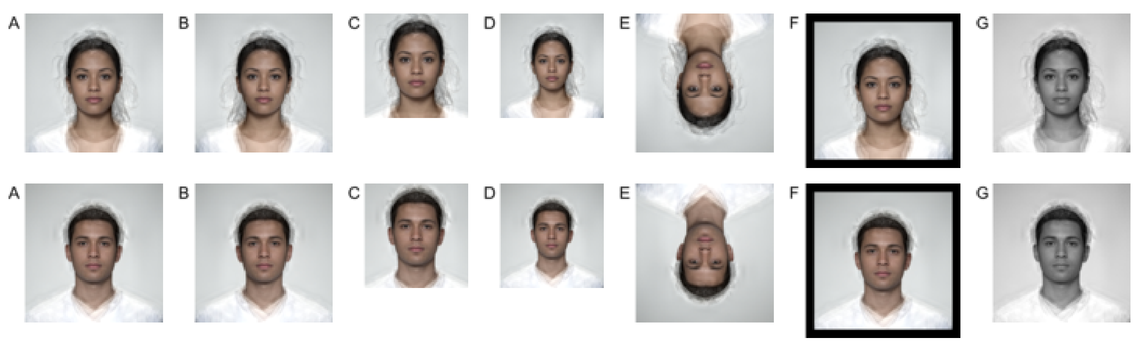

Figure 3. Examples of image manipulations: (A) original image, (B) mirrored, (C) cropped to 75%, (D) resized to 75%, (E) rotated 180 degrees, (F) 30 pixels of black padding added, and (G) greyscale.

Delineation. The image manipulations above work best if your raw images start the same size and aspect ratio, with the faces in the same orientation and position on each image. This is frequently not the case with raw images. Image delineation provides a way to set image manipulation parameters relative to face landmarks by marking corresponding points according to a template.

WebMorph.org's default face template marks 189 points (Fig. 4). Some of these points have very clear anatomical locations, such as point 0 ("left pupil"), while others have only

Table 2
The first 10 landmark points of WebMorph.org's default "FRL" template.

n	name	x	y	sym
0	left pupil	166	275	1
1	right pupil	284	275	0
2	top of left iris	165	267	10
3	top-left of left iris	156	270	17
4	left of left iris	154	277	16
5	bottom-left of left iris	157	283	15
6	bottom of left iris	166	286	14
7	bottom-right of left iris	174	283	13
8	right of left iris	177	276	12
9	top-right of left iris	175	270	11

approximate placements and are used mainly for masking or preventing morphing artifacts from affecting the background of images, such as point 147 ("about 2cm to the left of the top of the left ear (creates oval around head)"). Template point numbering is 0-based because PsychoMorph was originally written in Java.

The function `tem_def()` retrieves a template definition that includes point names, default coordinates, and the identity of the symmetrically matching point for mirroring or symmetrising images Table 2.

You can automatically delineate faces with a simpler template (Fig. 5) using the online services provided through the free web platform Face++ (2021), or dlib models provided by Davis King on a CC-0 license and included in the `webmorphR.dlib` package.

```
# load 5 images with FRL templates
f <- load_stim_neutral("006|038|064|066|135")

# remove templates and auto-delineate with dlib
# requires a python installation
dlib70_tem <- auto_delin(f, "dlib70", replace = TRUE)
dlib7_tem <- auto_delin(f, "dlib7", replace = TRUE)

# remove templates and auto-delineate with Face++
# requires a Face++ account; see ?webmorphR::auto_delin
fpp106_tem <- auto_delin(f, "fpp106", replace = TRUE)
fpp83_tem <- auto_delin(f, "fpp83", replace = TRUE)
```

A study comparing the accuracy of four common measures of face shape (sexual dimorphism, distinctiveness, bilateral asymmetry, and facial width to height ratio) between automatic and manual delineation concluded that automatic delineation had higher replicability and good correlations with manual delineation (A. L. Jones et al., 2021). However, around 2% of images had noticeably inaccurate automatic delineation, which should be screened for by outlier detection and visual inspection.

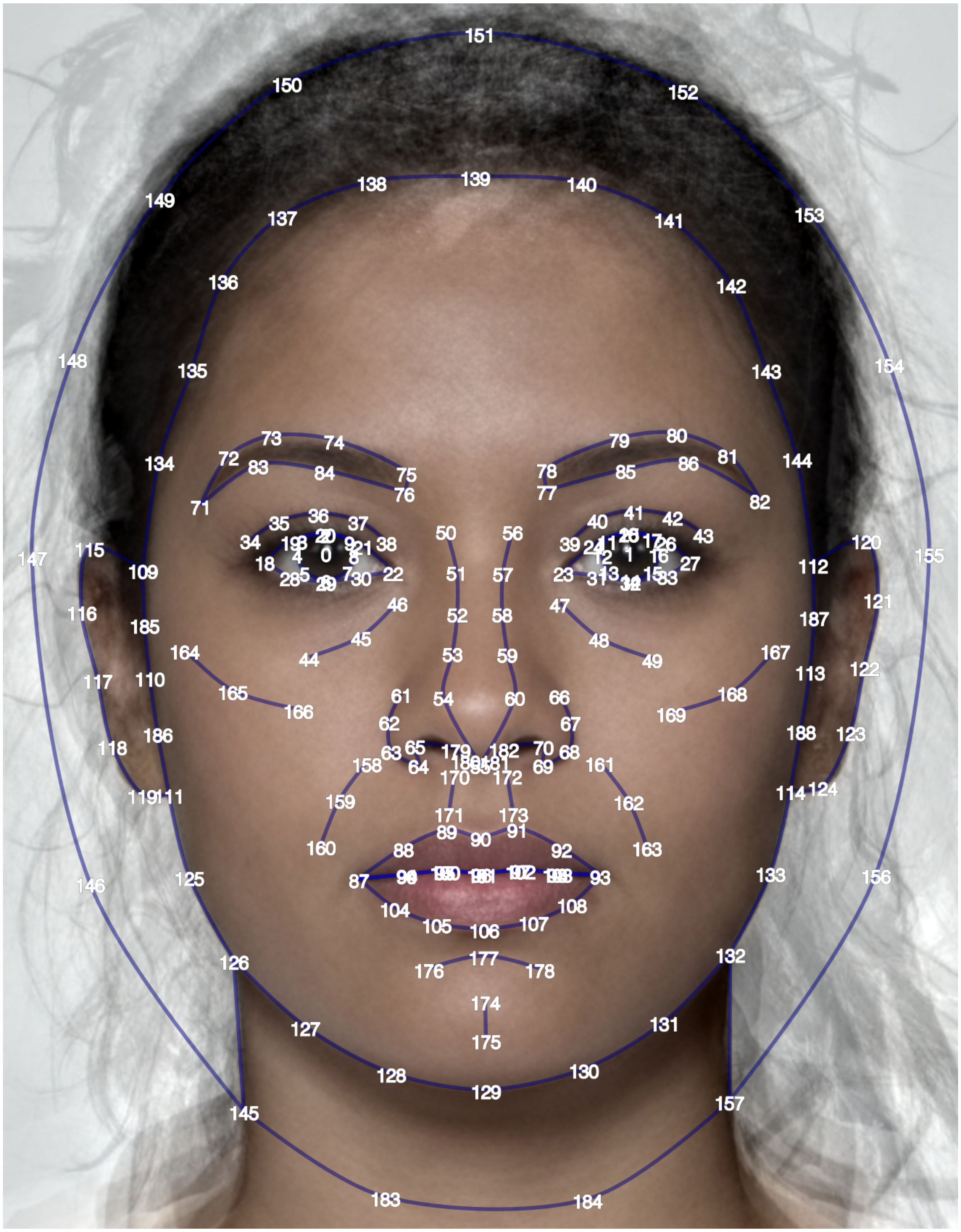

Figure 4. Default webmorph FRL template

Figure 5. Delineation templates: (A) manual delineation using the FRL template, (B) automatic delineation using the Face++ 106-point template, (C) automatic delineation using the Face++ 83-point template, (D) automatic delineation using the 70-point dlib template, and (E) automatic delineation using the 7-point dlib template.

Table 3
Coordinates of the first two points.

image	point	x	y
006_03	0	570	620
006_03	1	776	630
038_03	0	580	580
038_03	1	793	577
064_03	0	570	578
064_03	1	783	570
066_03	0	562	595
066_03	1	790	599
135_03	0	573	639
135_03	1	788	639

You can use the `delin()` function in webmorphR to open auto-delineated images in a visual editor to fix any inaccuracies.

```
dlib7_tem_fixed <- delin(dlib7_tem)
```

While automatic delineation has the advantage of being very fast and generally more replicable than manual delineation, it is more limited in the areas that can be described. Typically, automatic face detection algorithms outline the lower face shape and internal features of the face, but don't define the hairline, hair, neck, or ears. Manual delineation of these can greatly improve stimuli created through morphing or transforming (Fig. 7).

Facial Metrics. Once you have images delineated, you can use the x- and y-coordinates to calculate various facial-metric measurements (Table 4). Get all or a subset of points with the function `get_point()`. Remember, points are 0-based, so the first point (left pupil) is 0. This function returns a data table with one row for each point for each face.

```
eye_points <- get_point(f, pt = 0:1)
```

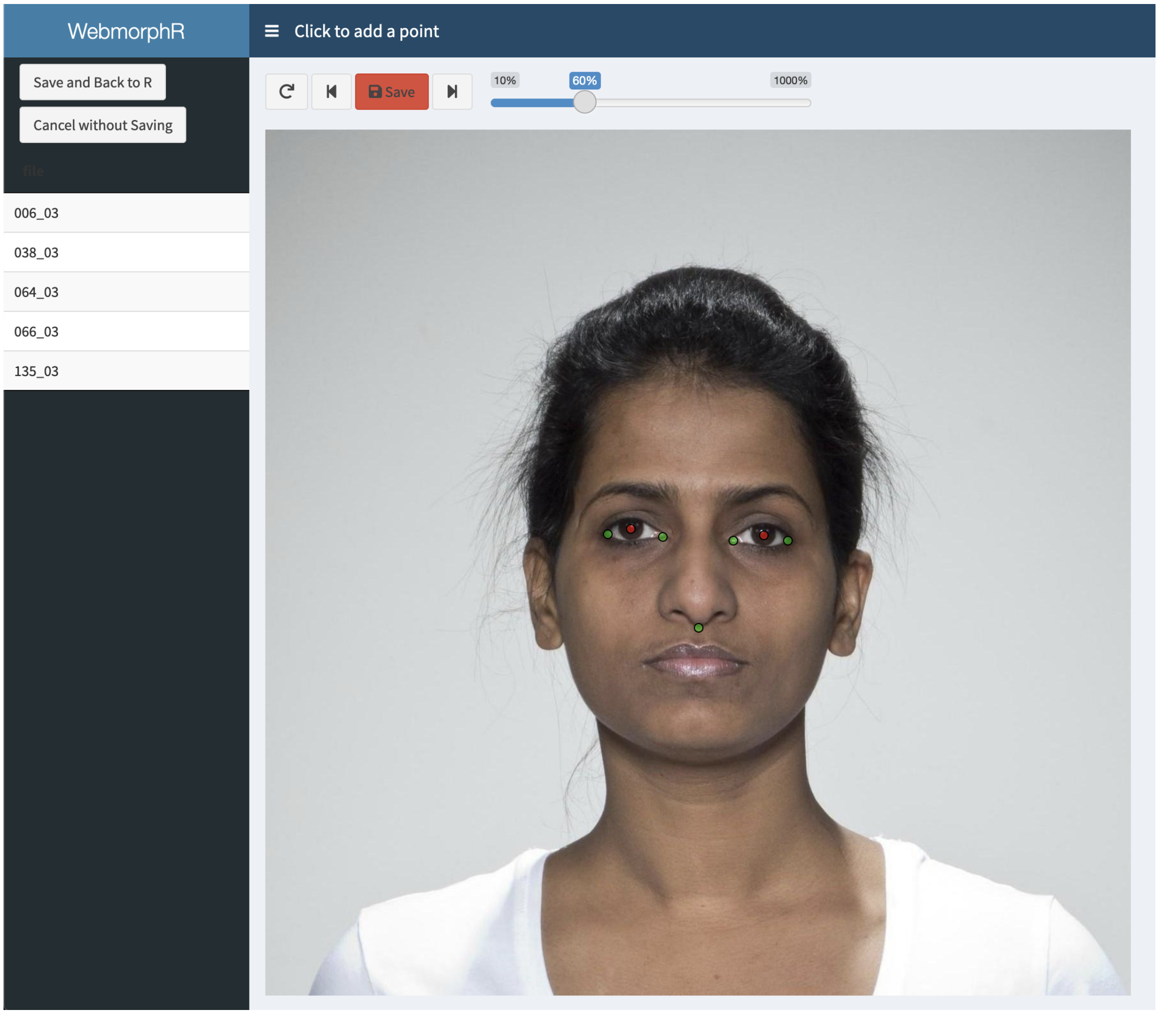

Figure 6. The shiny app interface for manual delineation adjustments.

The `metrics()` function helps you quickly calculate the distance between any two points, such as the pupil centres, or use a more complicated formula, such as the face width-to-height ratio from Lefevre et al. (2013).

```
# inter-pupillary distance between points 0 and 1
ipd <- metrics(f, c(0, 1))

# face width-to-height ratio
left_cheek <- metrics(f, "min(x[110],x[111],x[109])")
right_cheek <- metrics(f, "max(x[113],x[112],x[114])")
bizygomatic_width <- right_cheek - left_cheek
top_upper_lip <- metrics(f, "y[90]")
highest_eyelid <- metrics(f, "min(y[20],y[25])")
face_height <- top_upper_lip - highest_eyelid
fwh <- bizygomatic_width/face_height
```

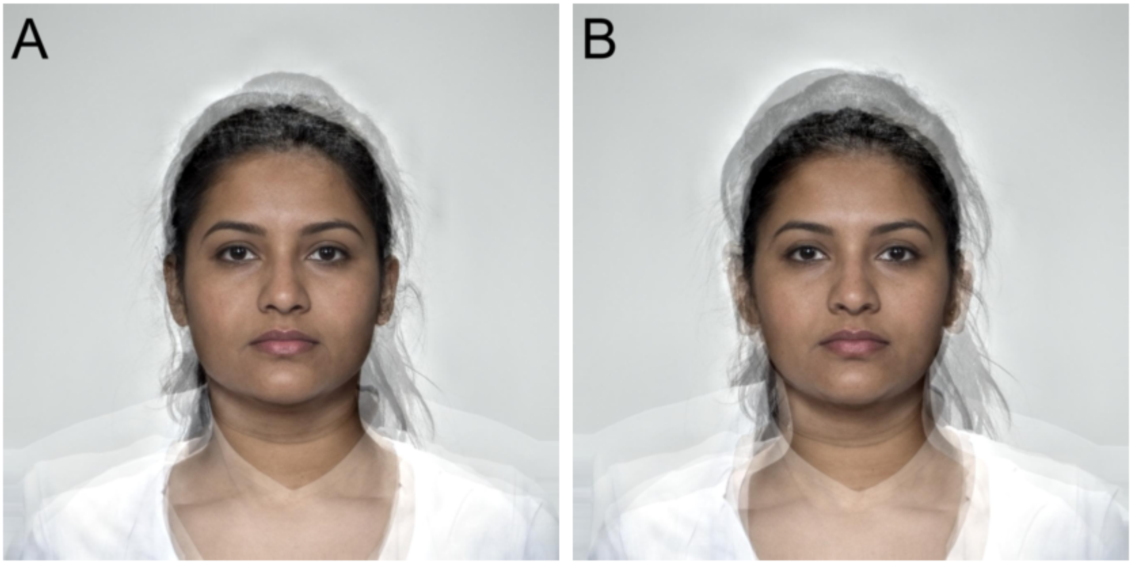

Figure 7. Averages of 5 images made using (A) the full 189-point manual template and (B) the reduced 106-point automatic template.

Table 4
Facial metric measurements.

face	x0	y0	x1	y1	ipd	fwh
006_03	570	620	776	630	206.2426	2.218905
038_03	580	580	793	577	213.0211	2.636580
064_03	570	578	783	570	213.1502	2.351220
066_03	562	595	790	599	228.0351	2.281818
135_03	573	639	788	639	215.0000	2.280788

```
# alternatively, do all calculations in one equation
fwh <- metrics(f, "abs(max(x[113],x[112],x[114])-min(x[110],x[111],x[109]))/abs(y[90]-min(y
```

While it is *possible* to calculate metrics such as width-to-height ratio from 2D face images, this does not mean it is a good idea. Even on highly standardized images, head tilt can have large effects on such measurements (Hehman et al., 2013). When image qualities such as camera type and head-to-camera distance are not standardized, facial metrics are meaningless at best (Trebicky et al., 2016).

Alignment. If your image set isn't highly standardised, you probably want to crop, resize and rotate your images to get them all in approximately the same orientation on images of the same size. There are several reproducible options, each with pros and cons.

One-point alignment (Fig. 8A) doesn't rotate or resize the image at all, but aligns one of the delineation points across images. This is ideal when you know that your camera-to-head distance and orientation was standard (or meaningfully different) across images and you want to preserve this in the stimuli, but you still need to get them all in the same position and image size.

Two-point alignment (Fig. 8B) resizes and rotates the images so that two points (usually the centres of the eyes) are in the same position on each image. This will alter relative head size such that people with very close-set eyes will appear to have larger heads than people with very wide-set eyes. This technique is good for getting images into the same orientation when you didn't have any control over image rotation and camera-to-head distance of the original photos.

Procrustes alignment (Fig. 8C) resizes and rotates the images so that each delineation point is as aligned as possible across all images. This can obscure meaningful differences in relative face size (e.g., a baby's face will be as large as an adult's), but can be superior to two-point alignment. While this requires that the whole face be delineated, you can use a minimal template such as a face outline or the Face++ auto-delineation to achieve good results.

You can very quickly delineate an image set with a custom template using the `delin()` function in webmorphR if auto-delineation doesn't provide suitable points.

```r
# one-point alignment
onept <- align(f, pt1 = 55, pt2 = 55,
               x1 = width(f)/2, y1 = height(f)/2,
               fill = "dodgerblue")

# two-point alignment
twopt <- align(f, pt1 = 0, pt2 = 1, fill = "dodgerblue")

# procrustes alignment
proc <- align(f, pt1 = 0, pt2 = 1, procrustes = TRUE, fill = "dodgerblue")
```

Masking. Oftentimes, researchers will want to remove the background, hair, and clothing from an image to avoid confounds. For example, the presence versus absence of hairstyle information can reverse preferences for masculine versus feminine male averages (L. M. DeBruine et al., 2006).

The "standard oval mask" has enjoyed widespread popularity because it is straightforward to add to images using programs like PhotoShop. WebmorphR's `mask_oval()` function allows you to set oval boundaries manually (Fig. 9A) or in relation to minimum and maximum template coordinates for each face (Fig. 9B) or across the full image set. An arguably better way to mask out hair, clothing and background from images is to crop around the curves defined by the template (Fig. 9C).

```r
# standard oval mask
bounds <- list(t = 200, r = 400, b = 300, l = 400)
oval <- mask_oval(f, bounds, fill = "dodgerblue")
```

Figure 8. Original images with different alignments. (A) One-point alignment placing the bottom of the nose point in the centre of the image. (B) Two-point alignment placing the eye centre points in the same position as the average image. (C) Procrustes alignment moved, rotated, and resized all images to most closely match the average face. A blue background was used to highlight the difference here, but normally a colour matching the image background would be used or the images would be cropped.

```r
# template-aware oval mask
oval_tem <- f |>
  subset_tem(features("gmm")) |> # remove external points
  mask_oval(fill = "dodgerblue") # oval boundaries to max and min template points

# template-aware mask
masked <- mask(f, c("face", "neck", "ears"), fill = "dodgerblue")
```

Averaging. Creating average images (also called composite or prototype images) through morphing can be a way to visualise the differences between groups (Burton et al., 2005), manipulate averageness (Little et al., 2011), or create prototypical faces for image transformations.

Averaging faces with texture (Tiddeman et al., 2005, 2001) makes composite images look more realistic (Fig. 10A). However, averages created without texture averaging look smoother and may be more appropriate for transforming color (Fig. 10B).

Figure 9. Images masked with (A) an oval defined by image coordinates, (B) an oval defined by the minimum and maximum x- and y-coordinates of template points, or (C) to include face, ears and neck.

```
avg_tex <- avg(f, texture = TRUE)
avg_notex <- avg(f, texture = FALSE)
```

Transforming. Transforming alters the appearance of one face by some proportion of the differences between two other faces. This technique is distinct from morphing. For example, you can transform a face in the dimension of sexual dimorphism by calculating the shape and color differences between a prototype female face (Fig. 11A) and a prototype male face (Fig. 11B). If you morph an individual female face with these images, you get faces that are halfway between the individual and prototype faces (Fig. 11C,D). However, if you transform the individual face by 50% of the prototype differences, you get feminised and masculinized versions of the individual face (Fig. 11E,F).

If, for example, the individual female face was more feminine than the average female face, morphing with the average female face produces an image that is *less* feminine than the original individual, while transforming along the male-female dimension produces and image that is always *more* feminine than the original. Morphing with a prototype also results in an image with increased averageness, while transforming maintains individually distinctive features.

Transforming also allows you to manipulate shape and colour independently (Fig. 12).

Symmetrising. Although a common technique (e.g., Mealey et al., 1999), left-left and right-right mirroring (Fig. 13) is not recommended for investigating perceptions of facial

Figure 10. An average of 5 faces created (A) with texture averaging and (B) without.

symmetry. This is because this method typically produces unnatural images for any face that isn't already perfectly symmetric. For example, if the nose does not lie in a perfectly straight line from the centre point between the eyes to the centre of the mouth, then one of the mirrored halves will have a much wider nose than the original face, while the the other half will have a much narrower nose than the original face. In extreme cases, one mirrored version can end up with three nostrils and the other with a single nostril.

A morph-based technique is a more realistic way to manipulate symmetry Paukner et al. (2017). It preserves the individual's characteristic feature shapes and avoids the problem of having to choose an axis of symmetry on a face that isn't perfectly symmetrical. In this method, the original face is mirror-reversed and each template point is re-labelled. The original and mirrored images are averaged together to create a perfectly symmetric version of the image that has the same feature widths as the original face (Fig. 14). You can also use this symmetric version to create asymmetric versions of the original face through transforming: exaggerating the differences between the original and the symmetric version.

```
sym_both <- symmetrize(f)
sym_shape <- symmetrize(f, color = 0)
sym_color <- symmetrize(f, shape = 0)
sym_anti <- symmetrize(f, shape = -1.0, color = 0)
```

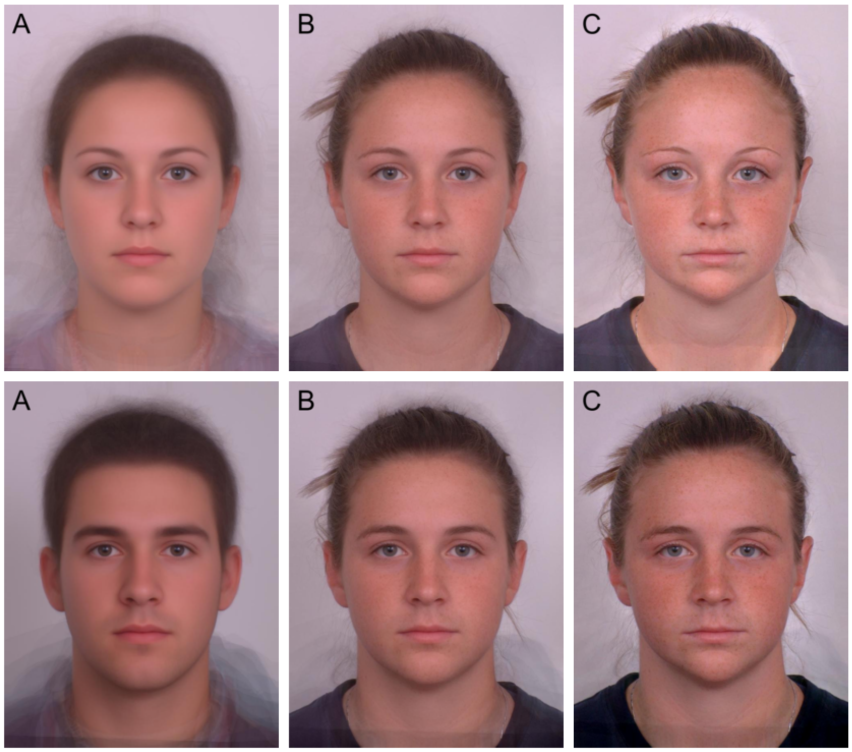

Figure 11. Morphing versus transforming: (A) female and male composite images, (B) averages of the composites with the individual image, (C) transforms of the individual image along the male-female continuum.

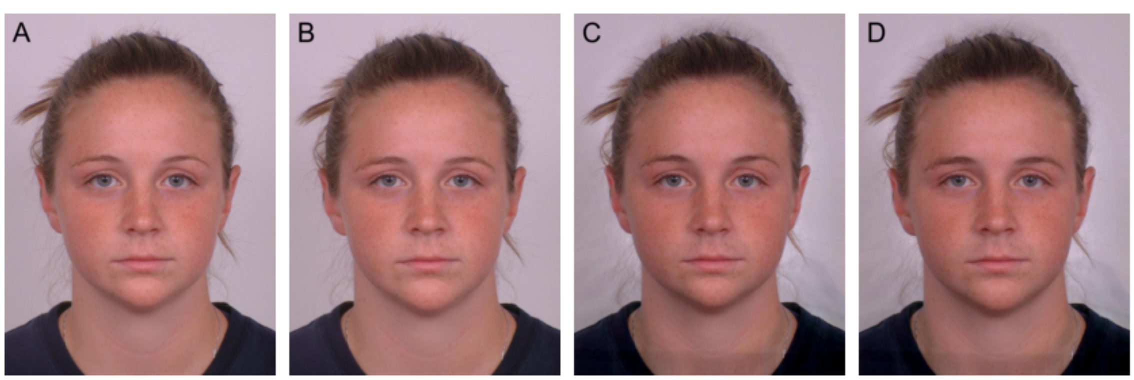

Figure 12. Transforming shape and color independently: (A) original individual image, (B) shape only, (C), color only, (D) both shape and color.

Figure 13. Left-left (top) and right-right (bottom) mirrored images. The code for making these images is in the supplemental materials, but we only recommend using this method to demonstrate how misleading it is.

Figure 14. Images with different types of symmetry: (A) symmetric shape and color, (B) symmetric color, (C) symmetric shape, (D) asymmetric shape.

Case Studies

In this section, we will demonstrate how more complex face image manipulations can be scripted, such as the creation of prototype faces, making emotion continuua, manipulating sexual dimorphism, manipulating resemblance, and labelling stimuli with words or images.

London Face Set. We will use the open-source, CC-BY licensed image set, the Face Research Lab London Set (L. M. DeBruine & Jones, 2017b). Images are of 102 adults whose pictures were taken in London, UK, in April 2012 for a project with Nikon camera (Fig. 15). All individuals were paid and gave signed consent for their images to be "used in lab-based and web-based studies in their original or altered forms and to illustrate research (e.g., in scientific journals, news media or presentations)."

Figure 15. The 102 neutral front faces in the London Face Set.

Each subject has one smiling and one neutral pose. For each pose, 5 full colour images were simultaneously taken from different angles: left profile, left three-quarter, front, right three-quarter, and right profile, but we will only use the front-facing images in the examples below. These images were cropped to 1350x1350 pixels and the faces were manually centered (many years ago before we made the tools in this paper). The neutral front images have template files that mark out 189 coordinates delineating face shape for use with Psychomorph or WebMorph.

Protoypes. The first step for many types of stimuli is to create prototype faces for some categories, such as expression or gender. The faces that make up these averages should be matched for other characteristics that you want to avoid confounding with the categories of interest, such as age or ethnicity. Here, we will choose 5 Black female faces, automatically delineate them, align the images, and create neutral and smiling prototypes (Fig. 16).

```
# select the relevant images and auto-delineate them
neu_orig <- subset(london, face_gender == "female") |>
  subset(face_eth == "black") |> subset(1:5) |>
  auto_delin("dlib70", replace = TRUE)
```

```
smi_orig <- subset(smiling, face_gender == "female") |>
  subset(face_eth == "black") |> subset(1:5) |>
  auto_delin("dlib70", replace = TRUE)

# align the images
all <- c(neu_orig, smi_orig)
aligned <- all |>
  align(procrustes = TRUE, fill = patch(all)) |>
  crop(.6, .8, y_off = 0.05)

neu <- subset(aligned, 1:5)
smi <- subset(aligned, 6:10)

neu_avg <- avg(neu, texture = FALSE)
smi_avg <- avg(smi, texture = FALSE)
```

Figure 16. Average and individual neutral and smiling faces.

We use the "dlib70" auto-delineation model, which is available through webmorphR.dlib (L. M. DeBruine, 2022b), but requires the installation of python and some python packages. However, it has the advantage of not requiring setting up an account at Face++ and doesn't transfer your images to a third party.

Emotion Continuum. Once you have two prototype images, you can set up a continuum that morphs between the images and even exaggerates beyond them (Fig. 17). Note that some exaggerations beyond the prototypes can produce impossible shape configurations, such as the negative smile, where the open lips from a smile go to closed at 0%

and pass through each other at negative values.

```
steps <- continuum(neu_avg, smi_avg, from = -0.5, to = 1.5, by = 0.25)
```

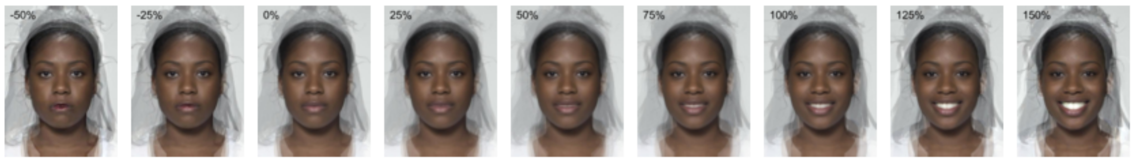

Figure 17. Continuum from -50% to +150% smiling.

Sexual dimorphism transform. We can use the full templates to create sexual dimorphism transforms from neutral faces. Repeat the process above for 5 male and 5 female neutral faces, skipping the auto-delineation because these images already have webmorph templates (Fig. 18).

```
# select the relevant images
f_orig <- subset(london, face_gender == "female") |>
  subset(face_eth == "black") |> subset(1:5)

m_orig <- subset(london, face_gender == "male") |>
  subset(face_eth == "black") |> subset(1:5)

# align the images
all <- c(f_orig, m_orig)
aligned <- all |>
  align(procrustes = TRUE, fill = patch(all)) |>
  crop(.6, .8, y_off = 0.05)

f <- subset(aligned, 1:5)
m <- subset(aligned, 6:10)

f_avg <- avg(f, texture = FALSE)
m_avg <- avg(m, texture = FALSE)
```

Next, transform each individual image using the average female and male faces as transform endpoints (Fig. 19).

```
# use a named vector for shape to automatically rename the images
sexdim <- trans(
  trans_img = c(f, m),
  from_img = f_avg,
  to_img = m_avg,
  shape = c(fem = -.5, masc = .5)
)
```

Figure 18. Average and individual female and male faces.

Self-resemblance transform. Some research involves creating "virtual siblings" for participants to test how they perceive and behave towards strangers with phenotypic kinship cues (L. M. DeBruine, 2004, 2005; L. M. DeBruine et al., 2011). As discussed in detail in DeBruine et al. (2008), while morphing techniques are sufficient to create same-gender virtual siblings, transforming techniques are required to make other-gender virtual siblings without confounding self-resemblance with androgyny (Fig. 20).

```
virtual_sis <- trans(
  trans_img = f_avg,    # transform an average female face
  shape = 0.5,          # by 50% of the shape differences
  from_img = m_avg,     # between an average male face
```

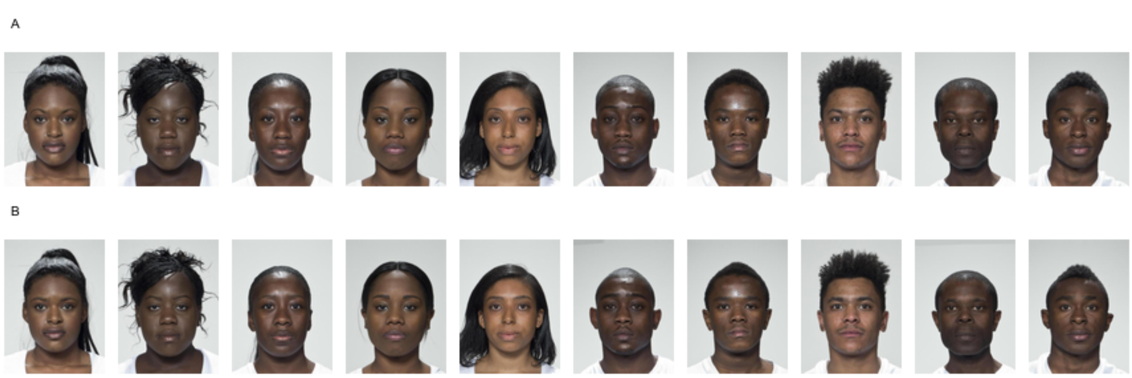

Figure 19. Versions of individual faces with (A) 50% feminised shape and (B) 50% masculinized shape.

```
  to_img = m) |>         # and individual male faces
  mask(c("face", "neck","ears"))

virtual_bro <- trans(
  trans_img = m_avg,     # transform an average male face
  shape = 0.5,           # by 50% of the shape differences
  from_img = m_avg,      # between an average male face
  to_img = m) |>         # and individual male faces
  mask(c("face", "neck","ears"))
```

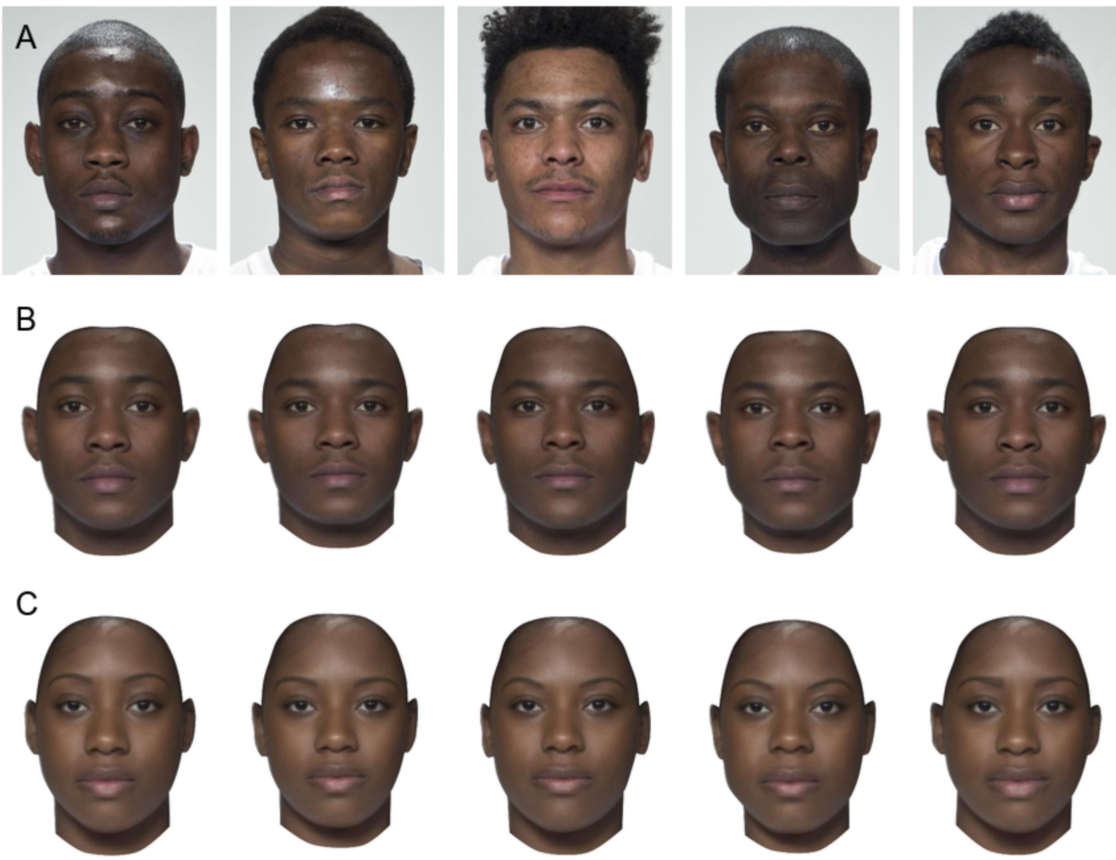

Figure 20. Creating virtual siblings: (A) original images, (B) virtual brothers, (C) virtual sisters.

Labels. Many social perception studies require labelled images, such a minimal group designs. You can add custom labels and superimpose images on stimuli (Fig. 21).

```
flags <- read_stim("images/flags")

ingroup <- f |>
  # pad 10% at the top with matching color
```

```
  pad(0.1, 0, 0, 0, fill = patch(f)) |>
  label("Scottish", "north", "+0+10") |>
  image_func("composite", flags$saltire$img,
             gravity = "northeast", offset = "+10+10")

outgroup <- f |>
  pad(0.1, 0, 0, 0, fill = patch(f)) |>
  label("Welsh", "north", "+0+10") |>
  image_func("composite", flags$ddraig$img,
            gravity = "northeast", offset = "+10+10")
```

Figure 21. Stimuli with text labels and superimposed images.

Discussion

Preparing your stimuli for face research in the ways described above has both personal and altruistic benefits. Once the original scripts are written, you will be able to prepare new stimuli without manual intervention. It also makes the process of changing your mind about the experimental design much less painful. If you decide that the images actually should have been aligned prior to several steps, you only need to add a line of code and rerun your script, instead of start a whole manual process over from scratch. But even more important, providing reproducible scripts can allow others to build on your work with their own images. This is beneficial for generalisability, whether or not you can share your original images.

In this section, we will discuss a number of issues related to making sure research that uses face stimuli is ethical and methodologically robust. While these issues may not be directly related to stimulus reproducibility, they are important to discuss in a paper that aims to make it easier for people to do research with face images.

Ethical Issues. Research with identifiable faces has a number of ethical issues. This means it is not always possible to share the exact images used in a study. In this case, it is all the more important for the stimulus construction methods to be clear and reproducible. However, there are other ethical issues outside of image sharing that we feel are important to highlight in a paper discussing the use of face images in research.

The use of face photographs must respect participant consent and personal data privacy. Images that are "freely" available on the internet, such as in Twitter profiles, are a grey area and the ethical issues should be carefully considered by the researchers and relevant ethics board.

We strongly advise against using face images in research where there is a possibility of real-world consequences for the pictured individuals. For example, do not post identifiable images of real people on real dating sites without the explicit consent of the pictured individuals for that specific research.

The use of face image analysis should never be used to predict behaviour or as automatic screening. For example, face images cannot be used to predict criminality or decide who should proceed to the interview stage in a job application. This type of application is unethical because the training data is always biased. Face image analysis can be useful for researching what aspects of face images give rise to the *perception* of traits like trustworthiness, but should not be confused with the ability to detect *actual* behaviour. Researchers have a responsibility to consider how their research may be misused in this manner.

Natural vs standardised source images. Use the right image for the question. – Ben, do you think you could write a bit about this? I thought it would be useful to explain when/why you might use standardised images versus naturalistic "holiday snaps". WebmorphR can help process either, but the delineations are mainly specialised for front-facing faces (although profile face templates are available).

```
left_profile <- tem_def(33)
right_profile <- tem_def(32)

left_viz <- viz_tem_def(left_profile)
right_viz <- viz_tem_def(right_profile)
```

Head position. Morphometrics – Iris, can you add this?

Judging composites. In this section we will explain a serious caveat to research using composite faces that concludes something about group differences from judgements of a single pair or a small number of pairs of composites. Since we are making it easier to create composites, we do not want to inadvertently encourage research with this particular design.

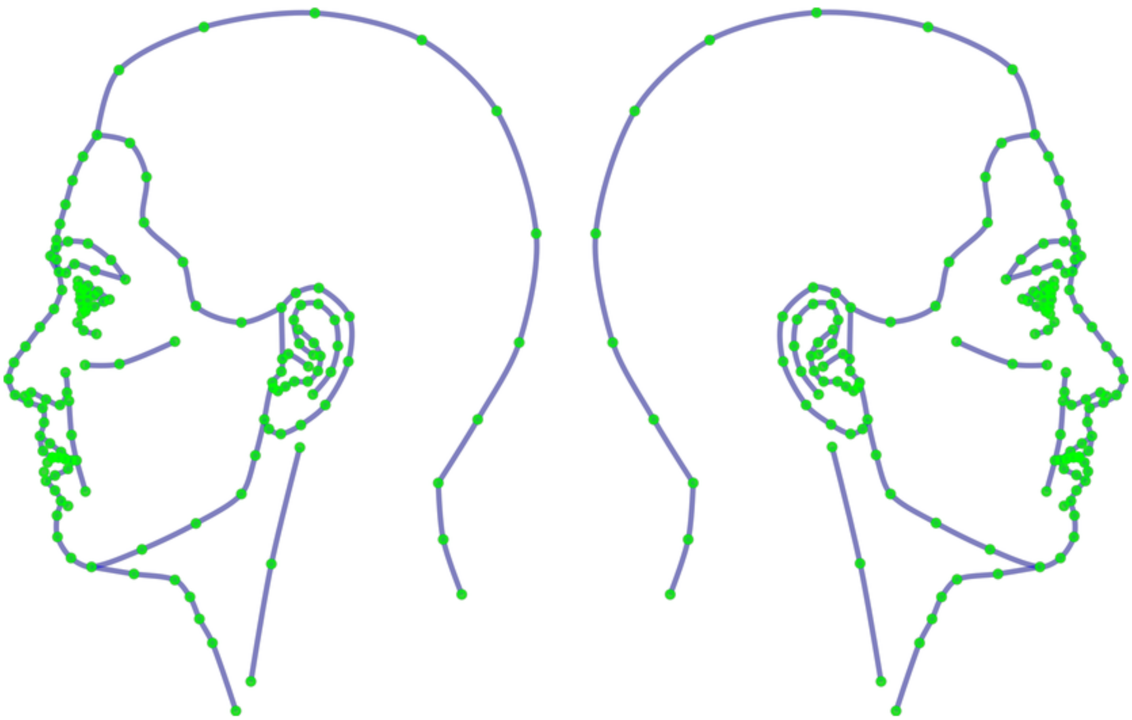

Figure 22. Left and right profile templates available via webmorph.org.

As a concrete illustration, a recent paper by Alper et al. (2021) used faces from the Faceaurus database (Holtzman, 2011b). "Holtzman (2011) standardized the assessment scores, computed average scores of self- and peer-reports, and ranked the face images based on the resulting scores. Then, prototypes for each of the personality dimensions were created by digitally combining 10 faces with the highest, and 10 faces with the lowest scores on the personality trait in question (Holtzman, 2011)." This was done separately for male and female faces.

Since scores on the three dark triad traits are positively correlated, the three pairs of composite faces are not independent. Indeed, Holtzman states that 5 individuals were in all three low composites for the male faces, while the overlap was less extreme in other cases. With 105 observers, Holtzman found that the ability to detect the composite higher in a dark triad trait was greater than chance.

While we commend both Holtzman and Alper, Bayrak, and Yilmaz for their transparency, data sharing, and material sharing, we argue that this test has an effective N of 2, not 105, and that further replications using these images, such as those done by Alper, Bayrak, and Yilmaz, regardless of number of observers or preregistered status, lend no further weight of evidence to the assertion that dark triad traits are visible in physical appearance.

To explain this, we'll use an analogy that has nothing to do with faces (bear with us). Imagine a researcher predicts that women born on odd days are taller than women born on even days. Ridiculous, right? So let's simulate some data assuming that isn't true. The code below samples 20 women from a population with a mean height of 158.1 cm and an SD of 5.7. Half are born on odd days and half on even days.

```
set.seed(8675309)

stim_n <- 10
height_m <- 158.1
height_sd <- 5.7

odd <- rnorm(stim_n, height_m, height_sd)
even <- rnorm(stim_n, height_m, height_sd)

t.test(odd, even)
```

```
##
##  Welch Two Sample t-test
##
## data:  odd and even
## t = 1.7942, df = 17.409, p-value = 0.09016
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.7673069  9.5977215
## sample estimates:
## mean of x mean of y
##  161.1587  156.7435
```

A t-test shows no significant difference, which is unsurprising. We simulated the data from the same distribution, so we know for sure there is no real difference here. Now we're going to average the height of the women with odd and even birthdays. So if we create a full-body composite of women born on odd days, she would be 161.2 cm tall, and a composite of women born on even days would be 156.7 cm tall.

If we ask 100 observers to look at these two composites, side-by-side, and judge which one looks taller, what do you imagine would happen? It's likely that nearly all of them would judge the odd-birthday composite as taller. But let's say that observers have to judge the composites independently, and they are pretty bad with height estimation, so their estimates for each composite have error with a standard deviation of 10 cm. We then compare their estimates for the odd-birthday composite with the estimate for the even-birthday composite in a paired-samples t-test.

```
obs_n <-100 # number of observers
error_sd <- 10 # observer error

# add the error to the composite mean heights
odd_estimates <- mean(odd) + rnorm(obs_n, 0, error_sd)
```

```
even_estimates <- mean(even) + rnorm(obs_n, 0, error_sd)

t.test(odd_estimates, even_estimates, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  odd_estimates and even_estimates
## t = 3.3962, df = 99, p-value = 0.0009848
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  1.902821 7.250747
## sample estimates:
## mean difference
##        4.576784
```

Now the women with odd birthdays are significantly taller than the women with even birthdays (p = 0.00). Or are they?

We can be sure that by chance alone, our two composites will be at least slightly different on any measure, even if they are drawn from identical populations. The smaller the number of stimuli that go into each composite, the larger the mean (unsigned) size of this difference. With only 10 stimuli per composite (like the Facesaurus composites), the mean unsigned effect size of the difference between composites from populations with no real difference is 0.35 (in units of SD of the original trait distribution). If our observers are accurate enough at perceiving this difference, or we run a very large number of observers, we are virtually guaranteed to find significant results every time. Additionally, there is a 50% chance that these results will be in the predicted direction, and this direction will be replicable across different samples of observers for the same image set.

So what does this mean for studies of the link between personality traits and facial appearance? The analogy with birth date and height holds. As long as there are facial morphologies that are even slightly consistently associated with the *perception* of a trait, then composites will not be identical in that morphology. Thus, even if that morphology is totally unassociated with the trait as measured by, e.g., personality scales or peer report (which is often the case), using the composite rating method will inflate the false positive rate for concluding a difference.

The smaller the number of stimuli that go into each composite, the greater the chance that they will be visibly different in morphology related to the judgement of interest, just by chance alone. The larger the number of observers or the better observers are at detecting small differences in this morphology, the more likely that "detection" will be significantly above chance. Repeating this with a new set of observers does not increase the amount of evidence you have for the association between the face morphology and the measured trait. You've only measured it once in one population of faces. If observers are your unit of analyses, you are making conclusions about whether the population of observers can detect the difference between your stimuli, you cannot generalise this to new stimulus sets.

So how should researchers test for differences in facial appearance between groups? Assessment of individual face images, combined with mixed effects models (L. M. DeBruine & Barr, 2021), can allow you to simultaneously account for variance in both observers and stimuli, avoiding the inflated false positives of the composite method (or aggregating ratings). People often use the composite method when they have too many images for any one observer to rate, but cross-classified mixed models can analyse data from counterbalanced trials or randomised subset allocation.

Another reason to use the composite rating method is when you are not ethically permitted to use individual faces in research, but are ethically permitted to use non-identifiable composite images. In this case, you can generate a large number of random composite pairs to construct the chance distribution. The equivalent to a p-value for this method is the proportion of the randomly paired composites that your target pair has a more extreme result than. While this method is too tedious to use when constructing composite faces manually, scripting allows you to automate such a task.

```
set.seed(8675309) # for reproducibility

# load 20 faces
f <- load_stim_canada("f") |> resize(0.5)

# set to the number of random pairs you want
n_pairs <- 5

# repeat this code n_pairs times
pairs <- lapply(1:n_pairs, function (i) {
  # sample a random 10:10 split
  rand1 <- sample(names(f), 10)
  rand2 <- setdiff(names(f), rand1)

  # create composite images
  comp1 <- avg(f[rand1])
  comp2 <- avg(f[rand2])

  # save images with paired names
  nm1 <- paste0("img_", i, "_a")
  nm2 <- paste0("img_", i, "_b")
  write_stim(comp1, dir = "images/composites", names = nm1)
  write_stim(comp2, dir = "images/composites", names = nm2)
})
```

Open Resources. In conclusion, we hope that this paper has convinced you that it is both possible and desirable to use scripting to prepare stimuli for face research. You can

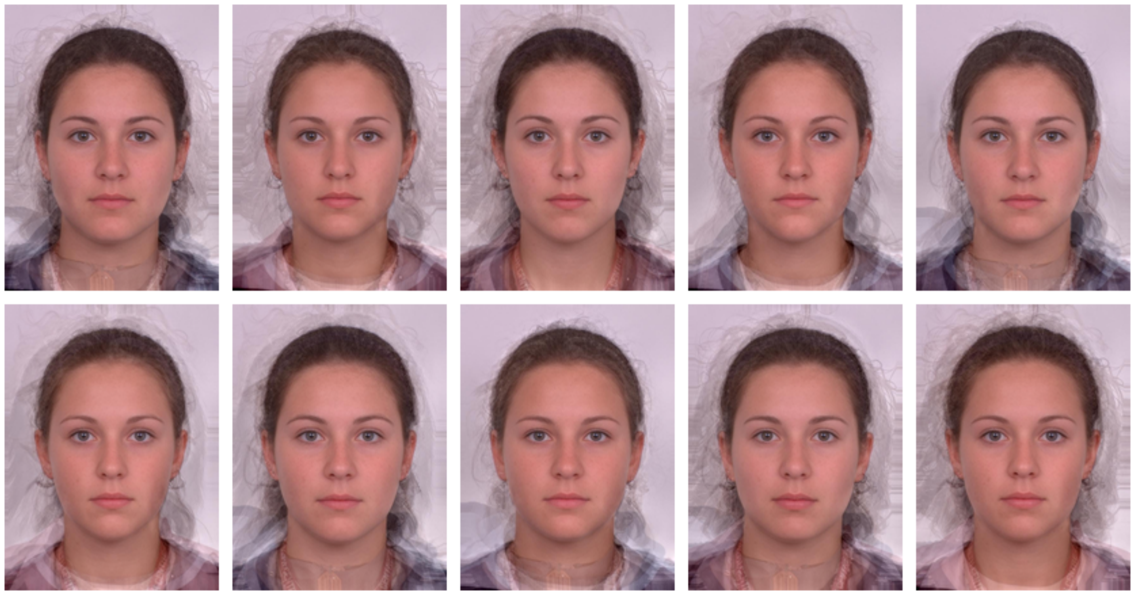

Figure 23. Five random pairs of composites from a sample of 20 faces (10 in each composite). Can you spot any differences?

access more detailed tutorials for webmorph.org at https://debruine.github.io/webmorph/ and for webmorphR at https://debruine.github.io/webmorphR/. All image sets used in this tutorial are available on a CC-BY license at figshare and all software is available open source. The code to reproduce this paper can be found at https://github.com/debruine/webmorphR/tree/master/paper.

References

We used R (Version 4.2.0; R Core Team, 2022) and the R-packages *dplyr* (Version 1.0.9; Wickham et al., 2022), *kableExtra* (Version 1.3.4; Zhu, 2021), *magick* (Version 2.7.3; Ooms, 2021), *papaja* (Version 0.1.0.9999; Aust & Barth, 2022), *tinylabels* (Version 0.2.3; Barth, 2022), *webmorphR* (Version 0.1.1.9001; L. M. DeBruine, 2022a, 2022b; L. DeBruine & Jones, 2017), *webmorphR.dlib* (Version 0.0.0.9003; L. M. DeBruine, 2022b), and *webmorphR.stim* (Version 0.0.0.9002; L. DeBruine & Jones, 2017) to produce this manuscript.

Alper, S., Bayrak, F., & Yilmaz, O. (2021). All the dark triad and some of the big five traits are visible in the face. *Personality and Individual Differences*, *168*, 110350. https://doi.org/https://doi.org/10.1016/j.paid.2020.110350

Aust, F., & Barth, M. (2022). *papaja: Prepare reproducible APA journal articles with R Markdown*. https://github.com/crsh/papaja

Barr, D. J. (2007). Generalizing over encounters. In *The oxford handbook of psycholinguistics*. Oxford University Press, USA.

Barth, M. (2022). *tinylabels: Lightweight variable labels*. https://cran.r-project.org/package=tinylabels

Benson, P. J., & Perrett, D. I. (1991a). Perception and recognition of photographic quality facial caricatures: Implications for the recognition of natural images. *European Journal of Cognitive Psychology*, *3*(1), 105–135.

Benson, P. J., & Perrett, D. I. (1991b). Synthesising continuous-tone caricatures. *Image and Vision Computing*, *9*(2), 123–129.

Benson, P. J., & Perrett, D. I. (1993). Extracting prototypical facial images from exemplars. *Perception*, *22*(3), 257–262.

Burton, A. M., Jenkins, R., Hancock, P. J., & White, D. (2005). Robust representations for face recognition: The power of averages. *Cognitive Psychology*, *51*(3), 256–284.

DeBruine, L. (2016). *Young adult composite faces*. figshare. https://doi.org/10.6084/m9.figshare.4055130.v1

DeBruine, L. isa M., Jones, B. C., Unger, L., Little, A. C., & Feinberg, D. R. (2007). Dissociating averageness and attractiveness: Attractive faces are not always average. *Journal of Experimental Psychology: Human Perception and Performance*, *33*, 1420–1430. https://doi.org/10.1037/0096-1523.33.6.1420

DeBruine, L. M. (2018). *Webmorph: Beta release 2* (Version v0.0.0.9001) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.1162670

DeBruine, L. M. (2004). Facial resemblance increases the attractiveness of same-sex faces more than other-sex faces. *Proceedings of the Royal Society of London B*, *271*, 2085–2090. https://doi.org/10.1098/rspb.2004.2824

DeBruine, L. M. (2005). Trustworthy but not lust-worthy: Context-specific effects of facial resemblance. *Proceedings of the Royal Society of London B*, *272*, 919–922. https://doi.org/10.1098/rspb.2004.3003

DeBruine, L. M. (2022a). *webmorphR : Reproducible stimuli*. Zenodo. https://doi.org/10.5281/zenodo.6570965

DeBruine, L. M. (2022b). *webmorphR.dlib : Face detection for webmorphR*. https://debruine.github.io/webmorphR.dlib/

DeBruine, L. M., & Barr, D. J. (2021). Understanding mixed-effects models through data simulation. *Advances in Methods and Practices in Psychological Science*, *4*(1), 2515245920965119.
DeBruine, L. M., & Jones, B. C. (2017a). *Young adult white faces with manipulated versions*. figshare. https://doi.org/10.6084/m9.figshare.4220517.v1
DeBruine, L. M., & Jones, B. C. (2017b). *Face research lab london set*. figshare. https://doi.org/10.6084/m9.figshare.5047666.v5
DeBruine, L. M., Jones, B. C., Little, A. C., Boothroyd, L. G., Perrett, D. I., Penton-Voak, I. S., Cooper, P. A., Penke, L., Feinberg, D. R., & Tiddeman, B. P. (2006). Correlated preferences for facial masculinity and ideal or actual partner's masculinity. *Proceedings of the Royal Society B: Biological Sciences*, *273*(1592), 1355–1360.
DeBruine, L. M., Jones, B. C., Little, A. C., & Perrett, D. I. (2008). Social perception of facial resemblance in humans. *Archives of Sexual Behavior*, *37*, 64–77. https://doi.org/10.1007/s10508-007-9266-0
DeBruine, L. M., Jones, B. C., Watkins, C. D., Roberts, S. C., Little, A. C., Smith, F. G., & Quist, M. (2011). Opposite-sex siblings decrease attraction, but not prosocial attributions, to self-resembling opposite-sex faces. *Proceedings of the National Academy of Sciences*, *108*, 11710–11714. https://doi.org/10.1073/pnas.1105919108
DeBruine, L., & Jones, B. (2017). *Face research lab london set*. figshare. https://doi.org/10.6084/m9.figshare.5047666.v5
DeBruine, L., & Jones, B. (2020). *3DSK face set with webmorph templates*. Open Science Framework. https://doi.org/10.17605/OSF.IO/A3947
Ekman, P. (1976). Pictures of facial affect. *Consulting Psychologists Press*.
Face++. (2021). Face++ AI open platform. In *Face++*. https://www.faceplusplus.com/landmarks/
Gonzalez, R. C., Woods, R. E.others. (2002). *Digital image processing*. Prentice Hall Upper Saddle River, NJ. https://www.pearson.com/us/higher-education/product/Gonzalez-Digital-Image-Processing-2nd-Edition/9780201180756.html
Gronenschild, E. H. B. M., Smeets, F., Vuurman, E. F. P. M., Boxtel, M. P. J. van, & Jolles, J. (2009). The use of faces as stimuli in neuroimaging and psychological experiments: A procedure to standardize stimulus features. *Behavior Research Methods*, *41*, 1053–1060. https://doi.org/10.3758/BRM.41.4.1053
Hehman, E., Leitner, J. B., & Gaertner, S. L. (2013). Enhancing static facial features increases intimidation. *Journal of Experimental Social Psychology*, *49*(4), 747–754. https://doi.org/10.1016/j.jesp.2013.02.015
Higham, D. J., & Higham, N. J. (2016). *MATLAB guide* (Vol. 150). Siam.
Holtzman, N. S. (2011a). Facing a psychopath: Detecting the dark triad from emotionally-neutral faces, using prototypes from the personality faceaurus. *Journal of Research in Personality*, *45*(6), 648–654.
Holtzman, N. S. (2011b). Facing a psychopath: Detecting the dark triad from emotionally-neutral faces, using prototypes from the personality faceaurus. *Journal of Research in Personality*, *45*(6), 648–654.
Holzleitner, I. J., Lee, A. J., Hahn, A. C., Kandrik, M., Bovet, J., Renoult, J.

P., Simmons, D., Garrod, O., DeBruine, L. M., & Jones, B. C. (2019). Comparing theory-driven and data-driven attractiveness models using images of real women's faces. *Journal of Experimental Psychology: Human Perception and Performance*, *45*(12), 1589.

Jones, A. L., & Jaeger, B. (2019). Biological bases of beauty revisited: The effect of symmetry, averageness, and sexual dimorphism on female facial attractiveness. *Symmetry*, *11*(2), 279.

Jones, A. L., Schild, C., & Jones, B. C. (2021). Facial metrics generated from manually and automatically placed image landmarks are highly correlated. *Evolution and Human Behavior*, *42*(3), 186–193. https://doi.org/10.1016/j.evolhumbehav.2020.09.002

Jones, B. C., Hahn, A. C., Fisher, C. I., Wang, H., Kandrik, M., Lao, J., Han, C., Lee, A. J., Holzleitner, I. J., & DeBruine, L. M. (2018). No compelling evidence that more physically attractive young adult women have higher estradiol or progesterone. *Psychoneuroendocrinology*, *98*, 1–5.

Lefevre, C. E., Lewis, G. J., Perrett, D. I., & Penke, L. (2013). Telling facial metrics: Facial width is associated with testosterone levels in men. *Evolution and Human Behavior*, *34*(4), 273–279.

Little, A. C., Burt, D. M., Penton-Voak, I. S., & Perrett, D. I. (2001). Self-perceived attractiveness influences human female preferences for sexual dimorphism and symmetry in male faces. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *268*(1462), 39–44.

Little, A. C., Jones, B. C., & DeBruine, L. M. (2011). Facial attractiveness: Evolutionary based research. *Philosophical Transactions of the Royal Society B*, *366*, 1638–1659. https://doi.org/10.1098/rstb.2010.0404

Ma, D. S., Correll, J., & Wittenbrink, B. (2015). The Chicago face database: A free stimulus set of faces and norming data. *Behavior Research Methods*, *47*, 1122–1135. https://doi.org/10.3758/s13428-014-0532-5

Mealey, L., Bridgstock, R., & Townsend, G. C. (1999). Symmetry and perceived facial attractiveness: A monozygotic co-twin comparison. *Journal of Personality and Social Psychology*, *76*(1), 151.

Morrison, D., Wang, H., Hahn, A. C., Jones, B. C., & DeBruine, L. M. (2018). *Predicting the reward value of faces and bodies from social perceptions: Supplemental materials*. OSF. https://doi.org/10.17605/OSF.IO/G27WF

Nishimura, D. (2000). GraphicConverter 3.9. 1. *Biotech Software & Internet Report: The Computer Software Journal for Scient*, *1*(6), 267–269.

Ooms, J. (2021). *Magick: Advanced graphics and image-processing in r*. https://CRAN.R-project.org/package=magick

Paluszek, M., & Thomas, S. (2019). Pattern recognition with deep learning. In *MATLAB machine learning recipes* (pp. 209–230). Springer.

Paukner, A., Wooddell, L. J., Lefevre, C. E., Lonsdorf, E., & Lonsdorf, E. (2017). Do capuchin monkeys (sapajus apella) prefer symmetrical face shapes? *Journal of Comparative Psychology*, *131*(1), 73.

Pegors, T. K., Mattar, M. G., Bryan, P. B., & Epstein, R. A. (2015). Simultaneous perceptual and response biases on sequential face attractiveness judgments.

Journal of Experimental Psychology: General, *144*(3), 664.
R Core Team. (2022). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing. https://www.R-project.org/
Rhodes, G. (2017). Adaptive coding and face recognition. *Current Directions in Psychological Science*, *26*(3), 218–224.
Rhodes, G., Yoshikawa, S., Clark, A., Lee, K., McKay, R., & Akamatsu, S. (2001). Attractiveness of facial averageness and symmetry in non-western cultures: In search of biologically based standards of beauty. *Perception*, *30*(5), 611–625. https://doi.org/10.1068/p3123
Rowland, D. A., & Perrett, D. I. (1995). Manipulating facial appearance through shape and color. *IEEE Computer Graphics and Applications*, *15*(5), 70–76.
Scheib, J. E., Gangestad, S. W., & Thornhill, R. (1999). Facial attractiveness, symmetry and cues of good genes. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *266*(1431), 1913–1917.
Sforza, A., Bufalari, I., Haggard, P., & Aglioti, S. M. (2010). My face in yours: Visuo-tactile facial stimulation influences sense of identity. *Social Neuroscience*, *5*(2), 148–162.
The ImageMagick Development Team. (2021). *ImageMagick* (Version 7.0.10) [Computer software]. https://imagemagick.org
Tiddeman, B. P., Burt, D. M., & Perrett, D. I. (2001). Prototyping and transforming facial textures for perception research. *IEEE Computer Graphics and Applications*, *21*(5), 42–50.
Tiddeman, B. P., Stirrat, M. R., & Perrett, D. I. (2005). Towards realism in facial image transformation: Results of a wavelet MRF method. *Computer Graphics Forum*, *24*, 449–456.
Trebicky, V., Fialova, J., Kleisner, K., & Havlicek, J. (2016). Focal length affects depicted shape and perception of facial images. *PLoS One*, *11*(2), e0149313.
Visconti di Oleggio Castello, M., Guntupalli, J. S., Yang, H., & Gobbini, M. I. (2014). Facilitated detection of social cues conveyed by familiar faces. *Frontiers in Human Neuroscience*, *8*, 678.
Wang, S.-Y., Wang, O., Owens, A., Zhang, R., & Efros, A. A. (2019). Detecting photoshopped faces by scripting photoshop. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10072–10081.
Weigelt, S., Koldewyn, K., & Kanwisher, N. (2013). Face recognition deficits in autism spectrum disorders are both domain specific and process specific. *PloS One*, *8*(9), e74541.
Wickham, H., François, R., Henry, L., & Müller, K. (2022). *Dplyr: A grammar of data manipulation.* https://CRAN.R-project.org/package=dplyr
Zhu, H. (2021). *kableExtra: Construct complex table with 'kable' and pipe syntax.* https://CRAN.R-project.org/package=kableExtra