Microsoft AI

Transparency Note

Azure Cognitive Services: Face API

Last Updated 3/29/19

Contents

About this Transparency Note

Accessible through Azure Cloud Services, Azure Face API ("Face API") detects, recognizes, and analyzes human faces in images using pre-trained machine learning models that have been developed by Microsoft. Developers can integrate Face API functions into their systems without creating their own models.

Facial recognition[1] is an important and useful technology that can improve efficiency, security, and customer experiences.

Face API is a building block for creating a facial recognition *system*. A facial recognition system includes the technology as well as the people who will use it, the people who will be subject to it, and the environment in which it is deployed. Creating a system that is fit for purpose requires an understanding of how the technology works, its capabilities and limitations, and how to achieve the most accurate results. This Transparency Note is intended to help you understand how Face API works, the choices you can make as a system owner that influence accuracy, and the importance of thinking about the whole system, including the technology, the people, and the environment. It is part of a broader effort at Microsoft to implement our Facial Recognition Principles, which set out how we approach the development and deployment of facial recognition technology. We encourage you to use the principles to guide your deployment efforts too.

The basics of Face API

Key facial recognition terms

Template	Images of people are converted to templates, which are then used for facial recognition. Machine-interpretable features are extracted from one or more images of an individual to create that individual's template. The images themselves – whether *enrollment* or *probe* images (see below) – are not stored by Face API and the original images cannot be reconstructed based on a template. Template quality is a key determinant of how accurate your results will be.
Enrollment	Enrollment is the process of enrolling images of individuals for template creation so they can be recognized. When a person is enrolled to a *verification* system used for authentication, their template is also associated with a primary identifier[2] that will be used to determine which template to compare with the probe template (see below). High-quality images and images representing natural variations in how a person looks (for instance wearing glasses, not wearing glasses) yield high-quality enrollment templates.

[1] This Transparency Note addresses Face API's recognition functions. Other functions, such as predicting attributes, including gender and age, and the limitations of these predictions are not addressed here.

[2] Face API does not store primary identifiers, such as customer IDs, alongside facial templates. Instead, Microsoft associates stored facial templates with random GUIDs or globally unique identifiers. System developers can associate the GUID generated by Microsoft with an individual's primary identifier to support verification of that individual.

Probe image	A probe image is an image submitted to a facial recognition system to be compared to enrolled individuals. Probe images are also converted to probe templates. As with enrollment templates, high-quality images result in high-quality templates.

Face API functions

Face API Detection ("Detection") answers the question, "*Are there one or more human faces in this image?*" Detection finds human faces in an image and returns bounding boxes indicating their locations. All other functions are dependent on Detection: before Face API can identify or verify a person (see below), it must know the locations of the faces to be recognized.

The Detection function can also be used to predict attributes about each face, including age and gender. These attribute prediction functions are completely separate from the verification and identification functions of Face API. Face API does not predict an individual's age or gender as a precursor to verifying or identifying them.

Face API can answer the questions:

1. Are there one or more human faces in this image?
2. Are these two images the same person?
3. Can this unknown person be matched to an enrolled template?

Face API Verification ("Verification") builds on Detection and addresses the question, *"Are these two images the same person?"*. In security or access scenarios, Verification relies on the existence of a primary identifier (such as a customer ID) and facial recognition is used as a second factor to *verify* the person's identity. Verification is also called "one-to-one" matching because the probe template (one person) is only compared to the template stored for the (one) person associated with the identification presented.

Face API Identification ("Identification) also starts with Detection and answers the question, "*Can this unknown person be matched to an enrolled template?* Identification compares a probe template to all enrollment templates stored in your private repository, so it is also called "one-to-many" matching. Candidate matches are returned based on how closely the probe template matches each of the enrolled templates.

Face API documentation

For more information on all of the functions of Face API, see the Face API documentation

Understanding accuracy and errors

How accurate is Face API?

Because Face API is a building block for creating a facial recognition system to which other building blocks must be added, it is not possible to provide a universally applicable estimate of accuracy for the actual system you are planning to deploy. Companies may share accuracy as measured by public benchmark competitions, but these accuracies depend on details of the benchmark and therefore won't be the same as the accuracy of a deployed system. Ultimately, system accuracy depends on a number of factors, including the technology and how it is configured, environmental conditions, the use case for the system, how people to be recognized interact with the camera, and how people interpret the system's output. The following section is intended to help you understand key concepts that describe accuracy in the context of a facial recognition system. With that understanding, we then describe system design choices and how they influence accuracy.

The language of accuracy

The **accuracy** of a facial recognition system is based on a combination of two things: how often the system correctly identifies a person who *is enrolled* in the system and how often the system correctly finds no match for a person who *is not enrolled*. These two conditions, which are referred to as the "true" conditions, combine with two "false" conditions to describe all possible outcomes of a facial recognition system:

True positive or true accept 	The person in the probe image is enrolled and they are correctly matched.
True negative or true reject 	The person in the probe image is not enrolled and they are not matched.
False positive or false accept 	Either the person in the probe image is not *enrolled* but is matched to an *enrolled* person OR the person in the probe image is enrolled but is matched with the wrong person.
False negative or false reject 	The person in the probe image is enrolled, but they are not matched.

The consequences of a false positive or a false negative vary depending on the purpose of the facial recognition system. The examples below illustrate this variation and how choices you make in designing the system affect the experience of those people who are subject to it.

Logging into a banking app

Facial recognition can provide an added layer of security in addition to a PIN or other primary identification. A false positive for this application reduces customer security because it results in an incorrect match, while a false negative could prevent the customer from accessing their account. Because the purpose of the system is security, false positives must be minimized and as a result, most errors will be false negatives (account access fails). To address this limitation, system owners can provide a fallback mechanism, like pushing a notification to the customer's phone with an access code. The customer's experience may be less efficient in this case, but account access is not blocked, and security is prioritized.

Organizing photographs

Many photo organizing apps help you find pictures of a specific person across your photo collection using facial recognition. In this instance, the customer is using the app to choose photos for a retirement party. Because the customer will be reviewing the photos and choosing photos they wish to use, false positives may not be particularly important: facial recognition is making the search task easier for the customer, and if they review a few more photos than necessary, they can still easily complete their task. On the other hand, if they have scanned old family photographs that are somewhat degraded, the app may not be able to find relatives in these photographs (false negatives) and the customer may be frustrated with the app.

These two examples illustrate that the impact of false positives and false negatives vary depending on the purpose of the system, and that the design of the whole system, including fallback mechanisms, determine the consequences for people when errors occur.

Match scores, match thresholds, and candidate lists

The purpose of this section is to help you understand how system configuration influences system accuracy and the trade-off between false positives and false negatives.

Match score	A match score describes the similarity between a probe template and an enrolled template. Match scores range from 0 to 1. High match scores indicate that it is more likely that the two images are of the same person.
Match threshold	A match threshold is a configurable value between 0 and 1 that determines the match score required to be considered a positive match. If the match threshold is set to 0, then any probe template will match any enrollment template. Face API has a default match threshold that you can change to suit your application.

When using the **Verification** function for authentication, if the match score between the probe template and the enrollment template associated with the primary identifier is at least as high as the match threshold, Face API will indicate that the probe image represents the person presenting identification.

When using the **Identification** function, it can be useful for a person to review a list of candidates ranked by match scores to determine the final match. Face API customers can choose how many candidate templates that reach the match threshold will be returned in ranked order of similarity to the probe template. These matches are referred to as a "**candidate list**". Face API will only return candidates with match scores at least as high as the match threshold. When no templates have match scores that reach the match threshold, no matches are returned.

Why choose a match threshold less than one?

Setting a match threshold allows for balancing the errors between false positives and false negatives to best address your specific scenario. The overall accuracy of the system is unlikely to be 100% and when the match threshold is set to 1, the strictest value, virtually all errors that occur will be false negatives: the system will return "no match" because the submitted probe template will not perfectly match any enrolled templates. Because match scores are affected by the quality of the probe and the enrollment images, a lower match score can indicate poor quality images, rather than less similarity between people in the images. When using Identification, if the match threshold is set too high, the system may not return enough candidates to find the true match. On the other hand, a low match threshold may return low quality matches and can reduce the efficiency and accuracy of the humans reviewing the matches.

How should a match threshold be selected?

The best match threshold for your system is based on:

- The system purpose
- The impact of false positives and false negatives on the people who will be subject to facial recognition
- Whether final judgments are made by a human
- How the whole system, including the experience design, supports resolution of errors.

Before selecting a match threshold, Microsoft recommends that you, as a facial recognition system owner, collect ground truth evaluation data on site to determine how the match threshold affects the achievement of your goals and affects people subject to and interpreting the output of the system.

Ground truth evaluation data is data that is collected to evaluate a system. Critically, the true identity for each person represented in a probe photo is known and can be correctly matched to their enrollment template. Ground truth labels can be compared to the output of the system to establish the overall accuracy and error rates, and the distribution of errors between false positives and false negatives. Ground truth evaluation data should include adequate sampling of diverse people who will be subject to recognition so that performance differences can be understood, and corrective action taken.

Based on the results of this evaluation, you can iteratively adjust the match threshold until the trade-off between false positives and false negatives meets your objectives.

Best practices for improving accuracy

Facial recognition technology is improving and many systems, including Microsoft's Face API, can perform well even when conditions are not ideal. However, these are specific actions that you can take to ensure best-quality results from your facial recognition system.

Plan for an evaluation phase

Before a large-scale deployment or rollout of any facial recognition system, Microsoft strongly recommends that system owners conduct an *evaluation phase* in the context where the system will be used and with the people who will interact with the system.

You should work with your analytics and research teams to collect ground truth evaluation data to:

- ☑ Establish baseline accuracy, false positive and false negative rates.
- ☑ Choose an appropriate match threshold to meet your objectives.
- ☑ Determine whether the error distribution is skewed towards specific groups of people.

This is likely to be an iterative process with adjustments to sensor position, lighting, and other factors that influence accuracy, as discussed in this section. This evaluation should reflect your deployment environment and any variations in that environment, such as lighting or sensor placement, as well as ground truth evaluation data that represents the diversity of people who will interact with your system.

In addition to telemetry data, you may also want to analyze feedback from the people making judgments based on the system output, satisfaction data from the people who are subject to recognition, and feedback from existing customer voice channels to help tune the system and ensure successful engagement.

Meet image quality specifications

Image quality is critical to quality facial recognition so you should ensure that both the images used to enroll people and the probe images meet the following specifications:

- ☑ Full-frontal head and shoulder view without obstruction.
- ☑ Face size is at least 200x200 pixels with at least 100 pixels between eyes. Faces are detectable when their size is as small as 36x36 pixels, but for best performance Microsoft recommends a minimum size of 200x200 pixels when using Face API.

☑ Enroll multiple images of each person. Include images that represent typical variations in how the person's face appears to the camera, for instance, with and without glasses, from varied angles.

Control image capture environment

Lighting and camera calibration

Pay attention to how well the detail of people's faces can be seen in images taken with the camera you are intending to use in the locations where you will use it. Face API uses RGB images.

☑ Capture images in appropriate lighting conditions. Is the lighting too bright, too dark? Are faces backlit? Is there too much light from one side and not enough from the other? When possible, place sensors away from areas with extreme lighting.

☑ Is the lighting adequate to accurately capture the details of people's faces with different skin tones?

Backgrounds

☑ Strive for neutral, non-reflective backgrounds. Avoid backgrounds containing faces, for instance where there are pictures of people displayed, or where people other than the person to be recognized are prominent in the photo.

Sensor placement and maintenance

☑ Position sensors at face-level to best capture images that meet the quality specifications.

☑ Ensure sensors are regularly checked for dust, smudges, and other obstructions.

Plan for variations in subject appearance and behavior

Facial occlusions

Facial recognition works best when the person's entire face is visible. Faces may be partially or entirely occluded for a variety of reasons, including:

- Religion: Headwear that covers or partially obscures faces.
- Weather: Garments like scarves wrapped across the face.
- Injury: Eye patches or large bandages.
- Vision Disability: Very opaque glasses and pinhole glasses (other glasses and lenses should be fine).
- Personal style: Bangs over eyebrows, baseball caps, large facial tattoos, etc.

It is not always possible to avoid occlusions: removing glasses may be unsafe and requiring removal of religious headwear may be impermissible. In addition to sensor placement, the following actions can help to address occlusion challenges:

☑ A fallback method, such as a non-biometric alternative, is critical. For some people, the fallback may be the option they consistently use.

☑ Pay attention to challenges that people face during evaluation and deployment to identify remediations that work best for your environment.

Abrupt changes in appearance

Dramatic changes in appearance, like the removal of a full beard or many years passing between enrollment and probe images (for adults), or even short periods of time between photos of children, can result in errors.

- ☑ In addition to supporting a fallback method, designing the user experience to support immediate re-enrollment following a recognition failure can improve user satisfaction.
- ☑ Facial recognition systems are generally less accurate for children. Microsoft recommends using Face API for recognition of people over 18. Facial recognition can be especially challenging with people 13 and younger.

Subject behavior

Image quality may be low when subjects are not facing the camera, occluding their face with their hands (such as brushing hair out of their eyes), moving too fast for the sensor to capture their image), or when their expression is extreme (like yawning widely with their eyes closed). To address these challenges:

- ☑ Design the user experience so people understand how to provide high-quality images.
- ☑ Create an environment where people naturally face the camera and slow down.
- ☑ Provide clear instructions for how people should behave during recognition (eyes open, mouth closed, stand still, etc.).

Biometric Twins

Twins, family members, and other people who look very similar to each other will be difficult for facial recognition systems to distinguish from one another. This is another reason to support a fallback method.

 Design the system to support human judgment

In most cases, Microsoft recommends using Face API's facial recognition capabilities to support people making more accurate and efficient judgements rather than fully automating a process. Meaningful human review is important to:

- Detect and resolve cases of misidentification or other failures.
- Provide support to people who believe their results were incorrect.
- Identify and resolve changes in accuracy due to changing conditions (like lighting or sensor cleanliness).

For example, when using Face API for building security, a trained security officer can help when the facial recognition fails to match someone who believes they are enrolled by deciding whether a person should be admitted to the building. In this case, Face API helps the security officer work more efficiently, requiring a judgment to admit someone only when the person is not recognized.

The user experience that you create to support the people who will use the system output should be designed and evaluated with those people to understand how well they can interpret the output, what additional information they might need, how they can get answers to their questions, and ultimately, how well the system supports their abilities to make more accurate judgments.

Face API supports facial recognition with still images: there are *no anti-spoofing countermeasures* built into Face API, such as depth or motion detection. In cases where facial recognition is supporting human judgment and improving efficiency, this is generally not a key limitation: humans can easily detect when a person is holding up a picture to a camera.

 Use multiple factors for authentication

Use Face API along with one or more other factors when creating authentication systems, such as confirming passengers who are about to board a plane or confirming a banking transaction. As discussed above, Verification makes use of facial recognition as a second factor for identifying someone rather than a single or

primary factor. Identification does not require another factor; however, Identification is a more technically difficult problem because the probe template is compared to ALL enrolled templates instead of just the template for the primary identifier associated with the probe template. It is often still possible to use other signals to support authentication when using Identification, such as narrowing the set of enrolled templates to compare by limiting the search to people who have a ticket for a specific flight. While it is not possible to choose Verification for all scenarios, Microsoft recommends Verification for uses including secure access to buildings and for other key business and security functions.

Deploying responsible facial recognition systems

In addition to addressing accuracy, here are some additional considerations for successful deployment.

Evaluate stakeholder concerns and design the experience to address them

Understand both the perceived value of the facial recognition system and the concerns that people may have about it. Engage your research team to help understand how your customers, employees, and other stakeholders can help you deploy a system that supports your critical needs and those of the people who will be involved.

Develop transparent communication and escalation processes for stakeholder concerns

People may still have questions and concerns. Part of any release plan should include both proactive and reactive communication, a documented escalation process, and clear explanations for how feedback will be addressed.

Provide training and evaluate the effectiveness of people who make final judgments based on facial recognition

Microsoft strongly recommends that customers develop training for people who will use the output of systems or who will decide whether the system output is correct. Customers should also evaluate whether these employees can make correct judgments based on the output of the system and determine whether any unfair biases are introduced.

Update privacy policies and implement necessary changes

Microsoft strongly recommends that private sector customers provide conspicuous notice to and secure consent from individuals before capturing their images for use with facial recognition technology. System owners should also establish responsible data handling practices (including limits on retention and reuse of images) and ensure that those practices are communicated clearly to individuals subject to the system. Remember to include considerations for children who may be subject to recognition. In some jurisdictions, there may be additional legal requirements, and customers are responsible for compliance with all applicable laws.

Learn more about Face API

Azure Cognitive Services Compliance and Privacy

Face API Technical Documentation

Learn more about how others are using Face API

Review Microsoft's Facial Recognition Principles

Contact us

Give us feedback on this document

Find out about support options

About this document

Last updated: 3/29/2019