

OneGroup - Automated Photo Sharing via Facial Recognition

Varun Agrawal
Georgia Institute of Technology
Atlanta, GA

April 12, 2016

Abstract

In recent years, Facial Recognition has improved in performance dramatically, leading to cloud based services such as Microsoft's Project Oxford. In this report, we demonstrate how we integrated Project Oxford with OneDrive and the Outlook Contacts API to create an automated photo-sharing feature, called OneGroup.

1 Introduction

Microsoft OneDrive provides an easy and intuitive way to backup and store files and data in the cloud. Most important are image files which get backed up from a user's device to the cloud via an automatic sync, thus safe-guarding the user's precious memories. However, it is usually the case that users have multiple people in the image and each person would like to see the image, prompting the user to login to OneDrive and manually share the images via email or a shareable link. We make two key observations. The first is that the people in the user's picture will most likely be in his contacts list since the user keeps communicating with them and is thus comfortable taking an image with them. The second is that the subset of the user's friend circle appearing in images with the user is a function of time, where newer pictures mostly have people who the user has recently met and is more likely to take more pictures with.

Using these two observations, we propose a feature called OneGroup.

2 Leveraging OneDrive and Outlook Contacts

In order to make the photo-sharing experience as seamless as possible, we leverage the backup synchronization capabilities of OneDrive. OneDrive provides webhooks for when a new file is created and we can use this webhook to initialize our feature.

When a user uploads a new photo to OneDrive, the image will be accessed via the OneDrive SDK and the content download URL can be retrieved for the image, which is then passed onto the Face API described below.

For sharing the image, once we identify the people in the image, if they are a part of the Person Group, the system can then automatically create a share link for the image and send the link to the person via their email ID.

3 The Face API

Microsoft's Project Oxford provides an easy and intuitive way to do face detection and recognition via HTTP endpoints. We first send the image (via its OneDrive URL) to the Face Detection API to find out all the faces present in the image.

Given the image and the list of bounding boxes for each face in the image, we can pass this information onto the Face Identification API to recognize who the person is. In order for the API to work, we need to define a PersonGroup which is a set of limited Persons whose representative face image is available to the API and it can train on. Using the PersonGroup API, we create a group which is then updated based on the Adaptive Weighting scheme described below. Initially, we will require the user to manually tag the images with the appropriate persons in order for the system to be able to create the PersonGroup, similar to how spam is marked in order for spam filters to learn the difference between spam and regular mail.

Once the PersonGroup is created, since the user tags the faces in the images using his contacts in Outlook, each Person in the PersonGroup is associated with an Outlook contact and thus sharing the image becomes trivial. We can simply set the Person ID in the PersonGroup set to be that person's email ID.

Examples of face detection and face identification are given in the accompanying images. We cannot provide more complete results due to privacy concerns and lack of adequate training data for creating the PersonGroup.

4 Adaptive Weighting of the PersonGroup

The second key feature of this system is its ability to dynamically update the PersonGroup. Since the number of Persons that can be created is limited to 1000, we use an adaptive weighing scheme for each Person in the PersonGroup.

The weighing scheme is quite straightforward. For each user, we initially set their weight to 1 and record the timestamp at which they were tagged by a user. Each time a person is tagged in the image, we increase their weight by 1, update their timestamp and normalize all the weights of the Persons in the PersonGroup such that the weights add up to 100. We use 100 in order to simplify the weight calculation and updates, though the weight can take any decimal value between 0 and 100.

If a new person is tagged in an image, we assign their weight to 1. We now need to eliminate one person from the PersonGroup due to the limit placed. Keeping in mind the observation that more recent images are likely to cluster amongst the same set of people, we find the Persons with the 5 lowest weights and delete the Person with the oldest timestamp. The lowest weights indicate that the person hasn't been tagged in a while and thus their weight has been driven down. To avoid eliminating Persons who were a part of a one-off picture and are likely to appear in a future picture, we eliminate the person with the

oldest timestamp, with the justification being that the person was already not being tagged in recent images and the user may have moved on to new social circles which are now dominating his images.

Initially, when this feature was conceived, there was a limit of 1000 persons per PersonGroup which can now be removed with a paid subscription. Keeping the original constraints in mind, we have developed OneGroup. However, as a premium feature, the user can pay to have the 1000 Person limit increased. This has the effect of extending the system to a wider set of Persons.

5 OneGroup Flow

We now go through the overall flow of OneGroup:

1. Initialize all Persons by performing face detection and asking the user to tag faces with their contacts in their Outlook Contacts.
2. Set each Person's weight to 1, record the timestamp of the person as the time at which the photo was taken in which they are tagged. Normalize the weights.
3. Add all the Persons to the PersonGroup with their Face images and train the PersonGroup
4. Each time a new image is uploaded to OneDrive, the webhook is called which starts OneGroup's functionality.
5. OneGroup performs face detection and face identification.
6. If a face is identified, the corresponding Person's weight and timestamp is updated and a link of the image is shared with that Person.
7. If a face is detected but not identified, we ask the user to enter the contact whom the face describes.

5.1 Conclusion

We have detailed the working of OneGroup and how it fits into the complete Microsoft cloud based eco-system. OneGroup leverages state-of-the-art capabilities in Photo storage, face detection, face identification and Microsoft's rich suite of APIs to provide a seamless and convenient way for users to quickly share group images with all the people present in the image.

Figure 1: Face Detection via Face API

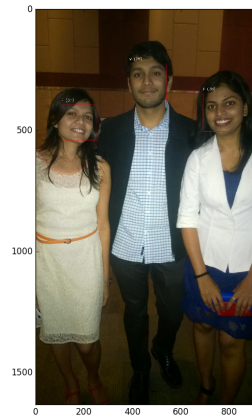


Figure 2: Face Identification via Face API

