

# Map User's Expressions into Emoji

Department of Electrical & Computer Engineering

Minghe Ren, Simin Zhai, Xueying Pan, Tianheng Hu

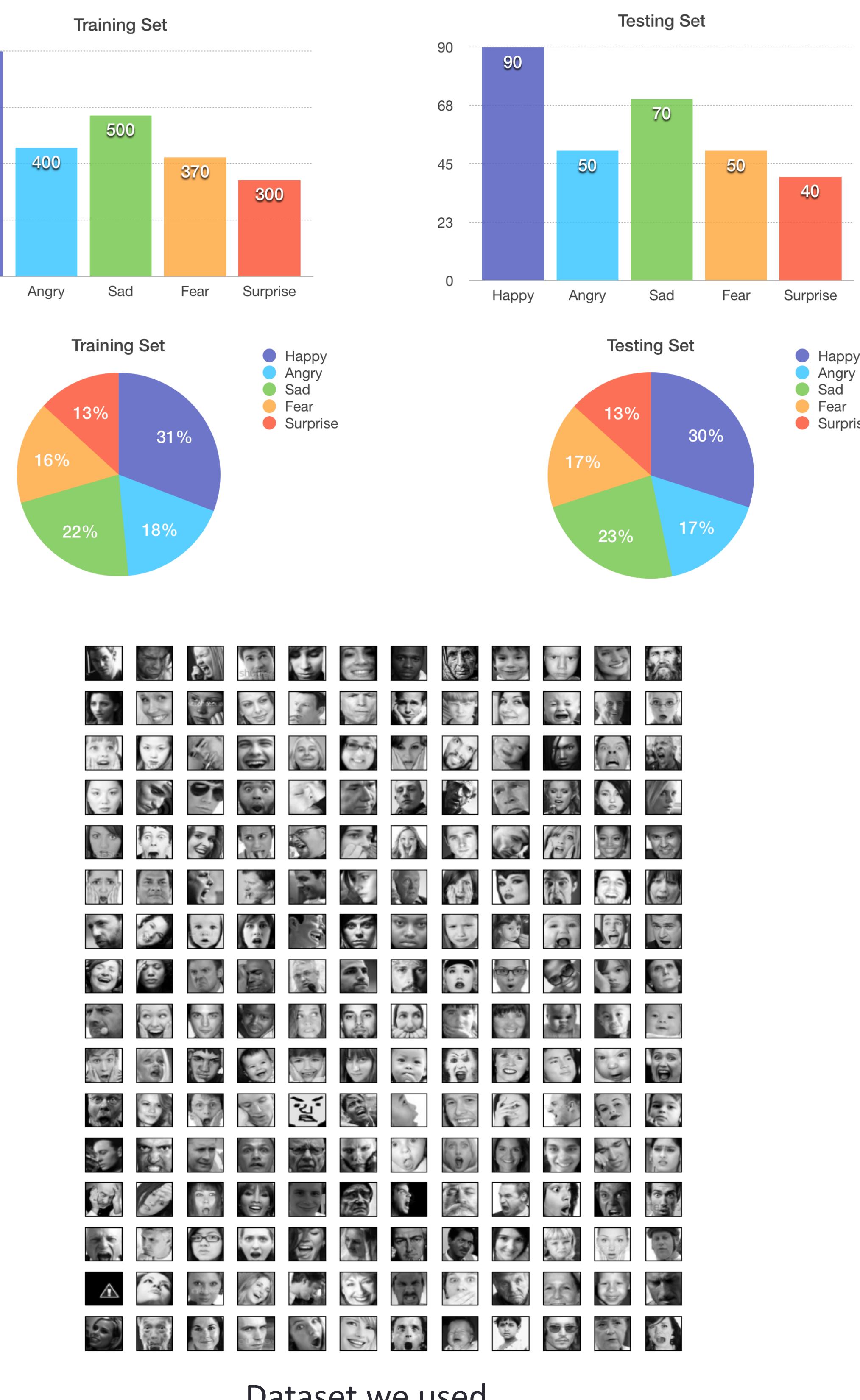
sawyermh@bu.edu, Siminz@bu.edu, xueying@bu.edu, tianheng@bu.edu

## Project Description

- Emoji Me is an android-based application to automatically attach your expression with cute emoji on your face. It is really fun to play because the mobile will understand your emotion, including sad, happy, angry, tedious, shock, fear, etc. You could also share and take the picture with your friends. We hope you will enjoy our magical application. Have a trip on our app now!

## Dataset

- We use open data source from Internet to train our model. Each picture is pre-cropped grey-scale human face with specific expression label. And we use a different data set as testing set. There are around 2770 pictures in training data set and 360 in testing data set. The final accuracy of our model is around 58%.

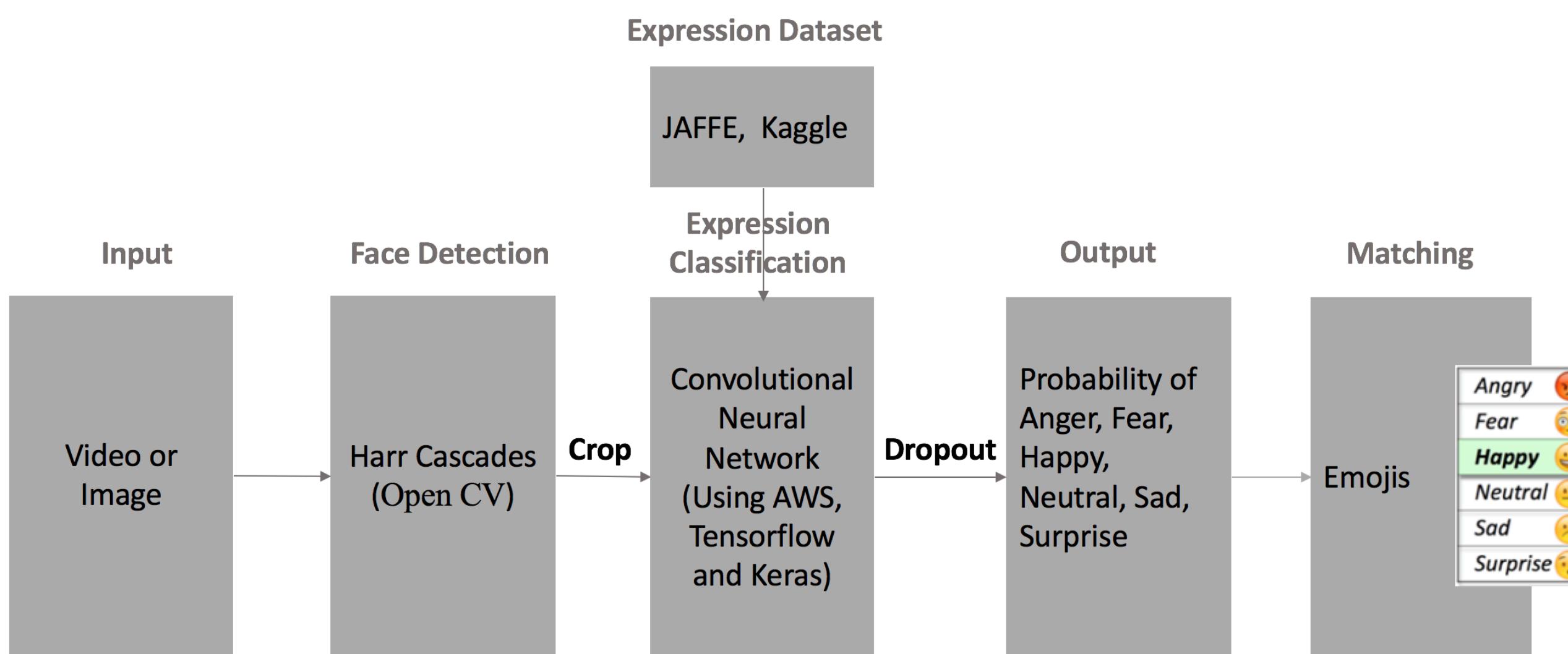


- Public test set[('angry', 52), ('fear', 49), ('happy', 89), ('sad', 65), ('surprise', 41), ('neutral', 60)]
- model accuracy: 57.34%

## Method

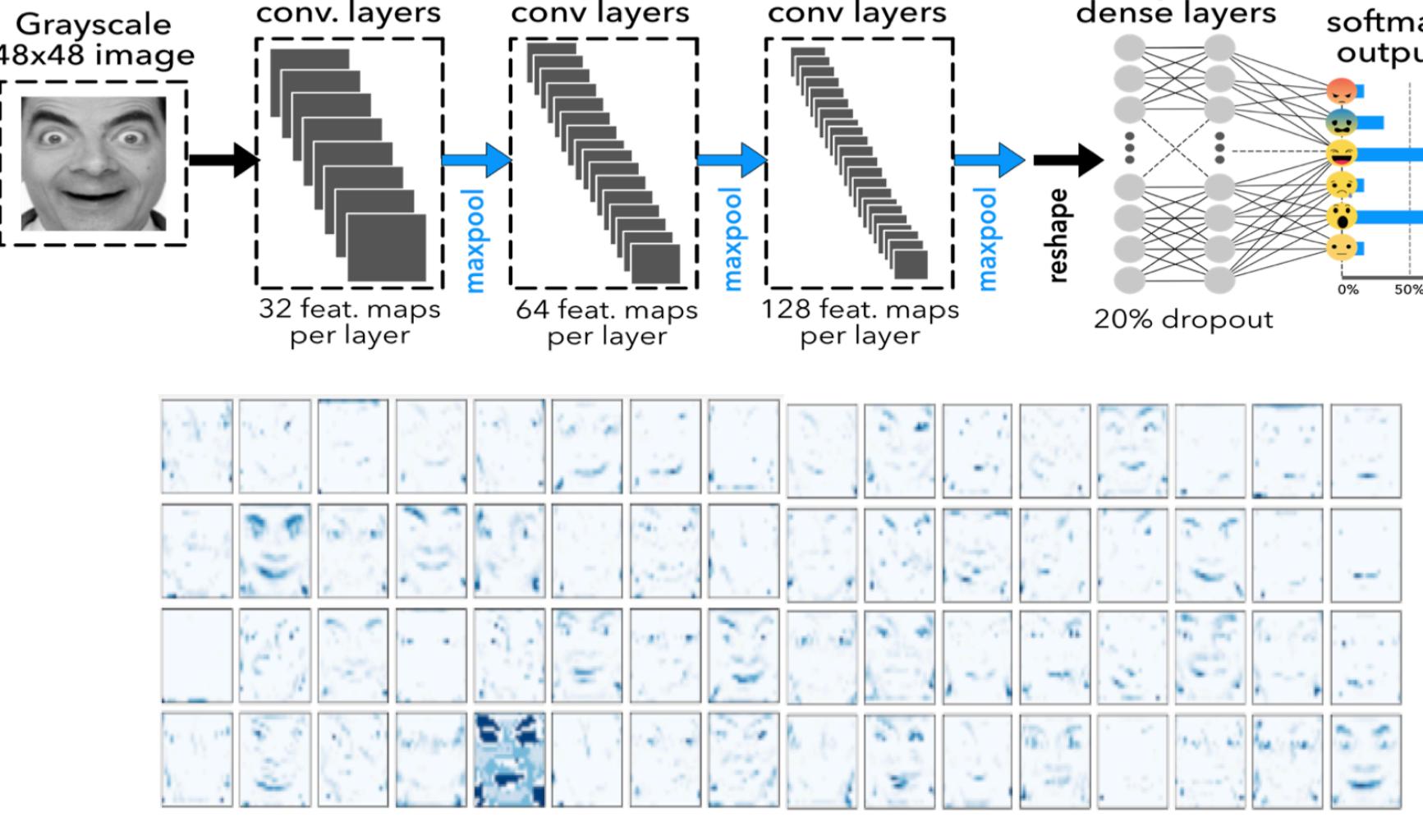
### 1. PC Model

- First, we use video or image as our input and use OpenCV Harr-Cascades model to crop out human face in a input. The Cropped input will be given into our CNN model and the output of our model would analyze the probability for each expression in this specific input. We will use the highest or the second highest probability of expression and match a proper emojis in our emojis library. Note that if the input is video, we will cut it into many frames and process them like pictures.



### 2. CNN Model

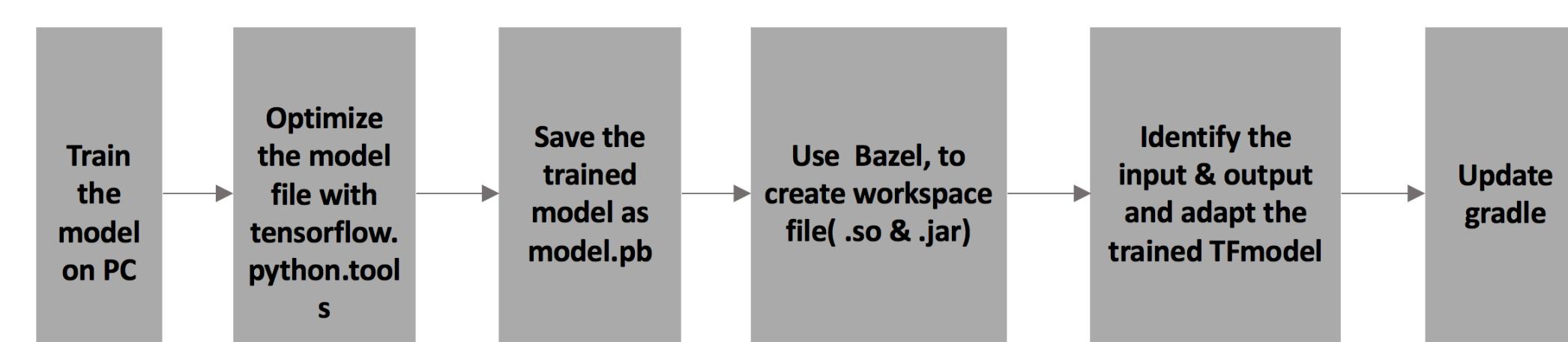
- Our CNN model is based on Keras using Tensorflow as backend. We trained our model in AWS for 12 hours. The model includes a input layer, 9 layers deep in convolution with one max-pooling after every three convolution layers, 2 dense layers (aka fully connected layers) and a softmax at the output layer to give the probability.



CNN (64-filter) feature maps after 2nd layer of max-pooling.

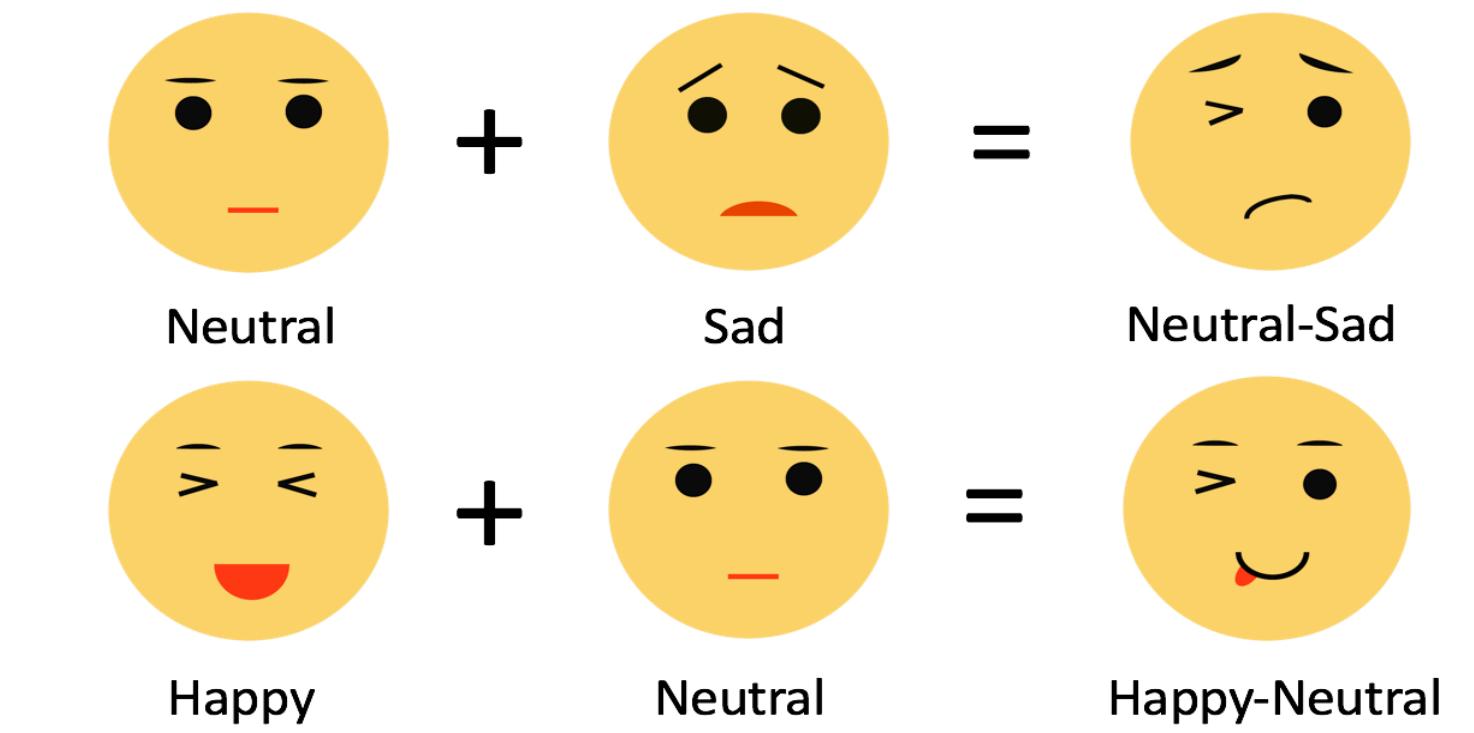
### 3. Import Model to Android

- The first step is to import the model file as model.pb. Then we add the tensorflow libraries to android studio and create a new application. The second step is to put the `model.py` file inside the assets folder and install Bazel, NDK, SDK and set up the workspace file. The final step is to put the so file and jar package of TF in the libs folder. Then we could call the model and enable tensorflow to run in the application.



## What's More

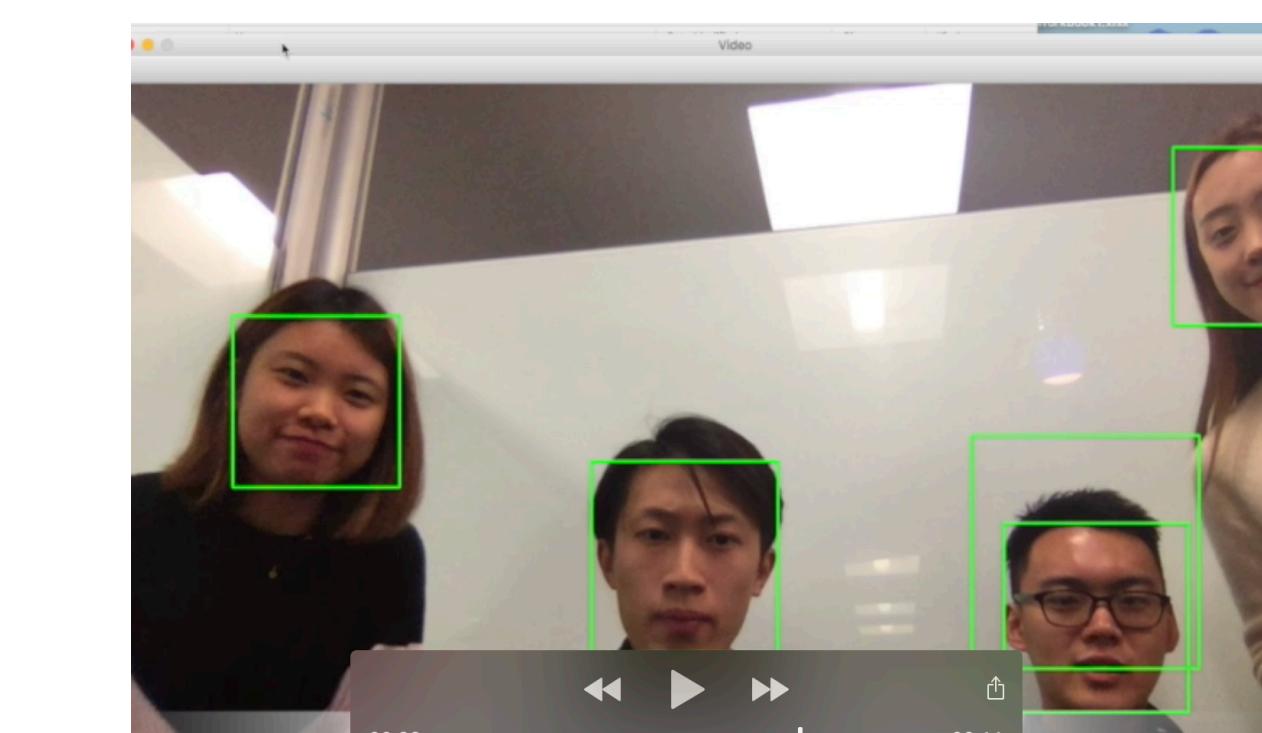
- We are trying to analyze more complicated expressions by using **combo**:
- Expression1 (**Highest Probability**) + Expression2 (**Second Highest Probability**) like:
- Neutral- Sad, Happy – Neutral... So we are trying to expand our emoji library.



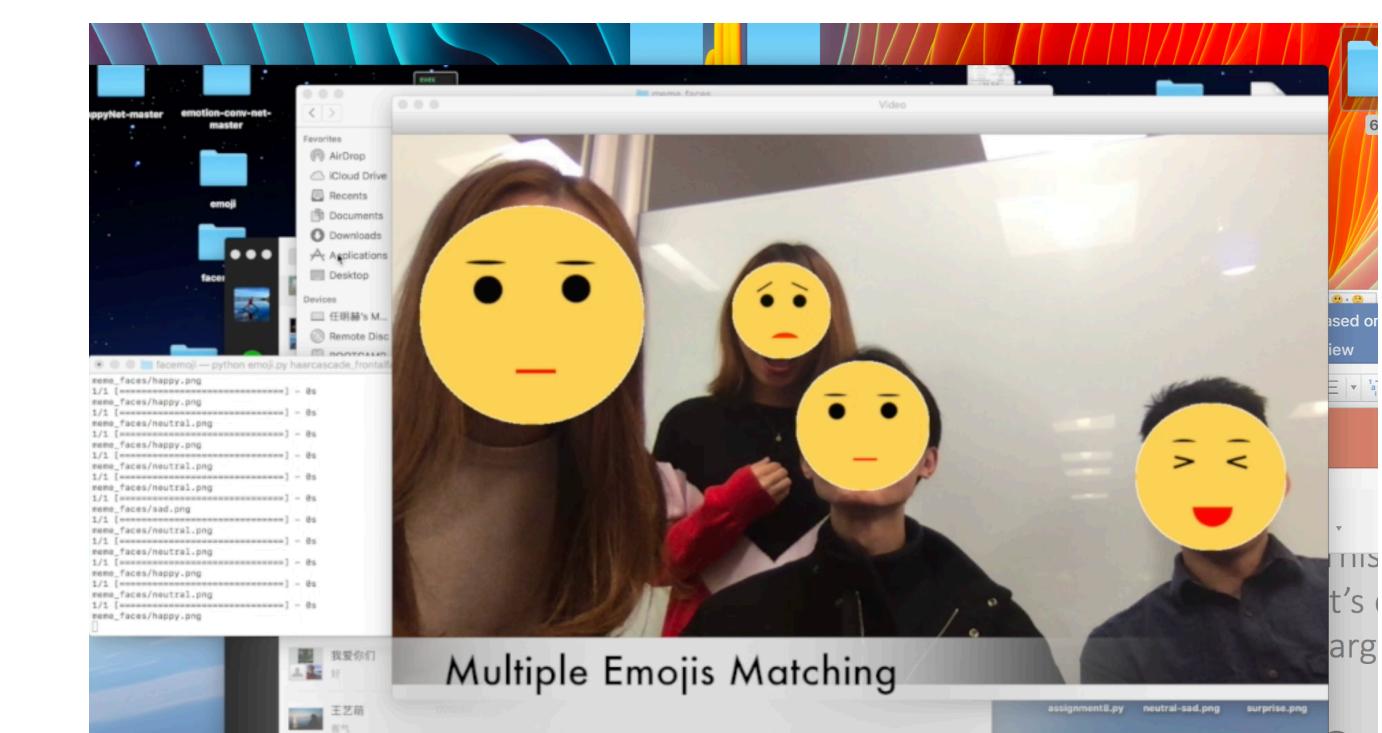
"2016 is the year when machines learn to grasp human emotions" -- Andrew Moore, the dean of computer science at Carnegie Mellon.

## Results

### 1. PC Demo:

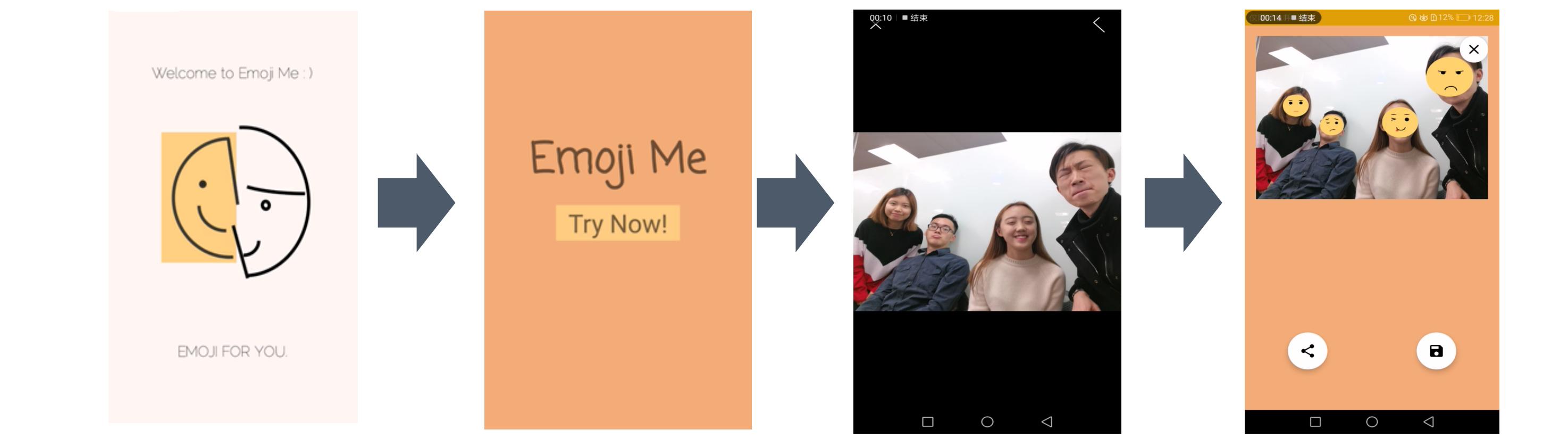


Face Detection



Emoji Matching

### 2. Android Demo:



## Accomplishments

- Creating 8 expressions, including 6 basic expressions (sad, happy, neutral, shock, fear, angry) and 2 advanced expressions (neutral-sad, neutral-happy)
- Implementing real-time video matching emojis on PC and photo matching emojis on android
- Detecting multiple people faces and matching their own emojis
- Tracing your face with emojis within 3 meters
- Friendly in the dark environment
- Save in album and Share your emoji photo with your social application