



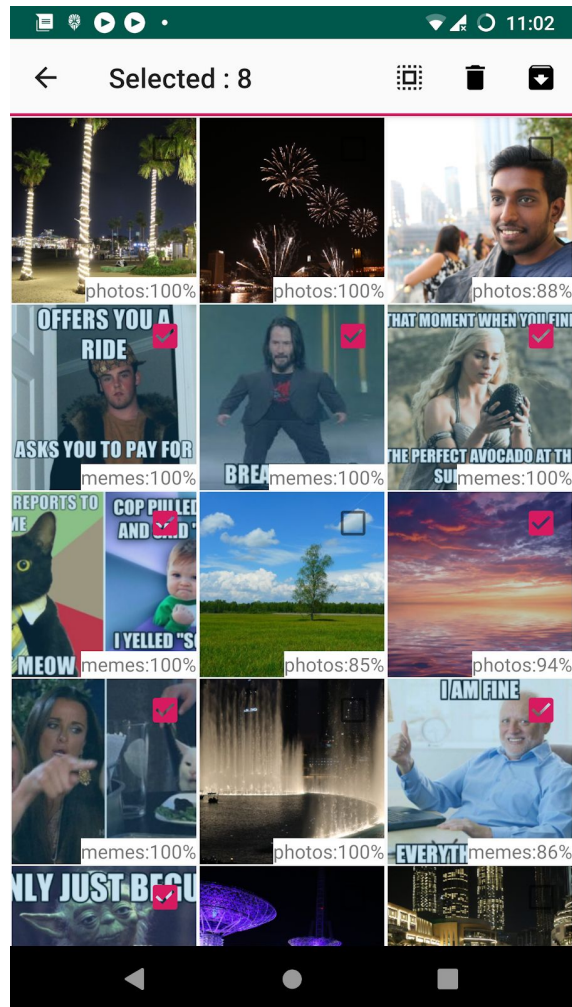
Tell us what your idea is.

Meme cleaner. The idea for the app came up to me when I was hit with a problem myself. My phone was running low on space, and when I checked the contents, I found that I had about 14K images on my Whatsapp Messenger images folder. When I went through those photos, I found that almost 80% of them were memes or something similar. Talking about it in my friend circle, the solution I got was to push it to Google Photos. However, I use Google Photos just for my camera roll so I didn't want to clutter my Google Photos gallery with it..

So my solution was to create an app that would help me cleanup the images on my phone. It would use machine learning to classify images into photos and memes, and help me in clearing out the clutter easily.

Tell us how you plan on bringing it to life.

I got hold of datasets for memes and photos and trained a Tensorflow model for classification. I converted it to tensorflow lite and made a working android prototype of it.



But I do realize a limitation here, that we **only** classify memes and images. Furthermore we only classify what the generic datasets had as memes and photos. So I have some ideas that could improve the app with On-Device ML.

- Feedback system that allows a user to mark a classification as wrong

The ML model could learn from the user's perspective of what are memes and what are photos. On-device ML is important here due to multiple reasons.

 - User privacy concern as the images can be private and the user may not want it to be uploaded to remote servers.
 - The ML model is customized to the user's requirements, storing a separate model for every user on a server would be impractical for scaling.
 - The time and bandwidth requirements for uploading data if feedback is given on large number of images.
- Capability to add additional categories as the user requires

Adding capability to add new categories, eg. screenshots, can widen the potential of this app.

I found that Tensorflow Lite's On-Device ML training is still on the roadmap and cloud based training and inference is impractical here as we would be uploading user's images. So an offline On-device model training solution can bring these ideas to life.

Existing timeline:

- November 30th, 2019 : Prototype with working classification
- December 15th, 2019 : Complete UI/UX design
- December 31st, 2019 : UI implementation
- January 15th, 2020 : Complete integration with UI
- January 31st, 2020 : Complete testing and release Version 1 to Google Play

Extended timeline with On-device ML:

- February, 2020 : Research On-device ML technologies
- February 29th, 2020 : On-device ML prototype
- March 15th, 2020 : UI design
- April 1st, 2020 : Complete implementation with UI
- April 15th, 2020 : Complete testing and release Version 2 to Google Play

Tell us about you.

I'm an under-graduate in Computer Science and Engineering from Mahatma Gandhi University, Kerala. I have developed a couple of Android apps during my college years. One of which was EventLane, an event listing app which helps users to locate nearby academic and professional events like seminars and conferences.

I got a full-time job during a campus drive from a product company which makes industry-leading enterprise level backup and recovery software solution. So I couldn't follow up with my projects after I joined. I'm still working on the same company for more than two years now and I got some free time now as I got familiarised with the professional ecosystem now. I started looking for more challenges during my free time.

So I started with Machine Learning and was looking around for a useful application for it. I found this idea and it was interesting and very helpful to me as well as my friends. I came across this Dev Challenge Program when I was going through the Android Documentation while developing the prototype.