Tell us what your idea is.

Recent years have seen a significant push towards mobile photography with phones like Pixels constantly pushing the limits of mobile camera hardware and software.
We believe that with the advent in Camera Technology, the size of images is bound to increase over time, and while services like Google Photos can help people not worry about the storage, that increases the complexity when it comes to organizing one's pictures.

With AfterShoot, we aim to develop a second pair of eyes that's been trained on pictures captured by a variety of sources ranging from professionals to hobbyist photographers to identify a bad picture and how it can be fixed.

Think of AfterShoot as your personal photographer who not only analyzes all the shots you take to weed out the bad ones but also points out how those shots can be improved.
Google's services we plan on using :

1. AutoML Vision Edge for training models to detect bad picture
2. Tensorflow to train the model to take parameters for a bad picture and suggest potential fixes
3. Firebase MLKit to serve these models and run them on the device

Apart from this, we also plan on using various photography techniques (Rule of Thirds, Rule of Depth and the Golden ratio to name a few) to identify an aesthetically pleasing image.

Aftershoot can also scan the photos in your gallery to help filter out potentially bad shots and help you manage your storage efficiently.

Note : All the image processing happens locally, so the user's privacy isn't violated.

Tell us how you plan on bringing it to life.

We're currently working on the models to identify and filter out good shots from the bad ones and we've so far :

1. Built the model that identifies blurry, overexposed, underexposed and best shot from duplicate shots.

2. Made a skeleton camera app (built upon the open sourced AOSP camera) which will be using the models we have above.

Google has already worked on identifying good shots from bad shots with Google Clips and we believe that the learning that Google had from Google Clips can help us accelerate the development of this project.

- (blurry, blinks, duplicates, overexposed, underexposed, sad faces, foreign object, cropped faces, etc.)

Our timeline for the project is :

- **November - December 2019** : Complete working on the models that identify potentially bad pictures :
  - Blurry/Out of Focus - **Done**
  - Blinks - **Done**
  - Over/Underexposed - **Done**
  - Duplicates, Sad Faces, Cropped Faces, Foreign Objects in the picture - **Work in Progress**
- **January - February 2019** : Work on a model that will take the flaw in the image (determined from the models above) and apply or suggest a fix that can be made to improve that image.
- **March - April 2019** : Finish work on the camera app and integrate it with the ML models
- **May 2019** : Beta test of the first version of the app

Most of the source code we have is for training the models above, so we won't be able to share it at the moment.

Source code for the camera, however, will be made public before the beta release of the app.

Tell us about you.

Android Developer and an avid tech blogger, I'm passionate about anything and everything related to Android.

Currently a Google Developer Expert for Firebase; I am also one of the first Google-certified Android developers in India.

Being an Open Source enthusiast, I've been a regular part of programs like Google Summer of Code and Google Code-In, both as a Mentor and a student.

Recently I have been inclined towards my new found love that combines my knowledge of Mobile Development with Machine Learning to create smart mobile apps.

Android Projects that I have worked on in the past:

1. offix-android : Worked as a maintainer of this open sourced SDK that provided offline support and Conflict Resolution for people using GraphQL in their Android Apps.
2. Open Event Android : Worked on an Android App Generator project that helped Event Organizers to generate Android and Web apps for their event with a single click
3. Pokidex : Android app that identifies Pokemon from the captured image, a real life Pokedex
4. CrapCons : An icon pack with a twist! CrapCons was the first icon pack for Android with all its assets made entirely in MS Paint.
5. ZooperCrap : Widget extension for CrapCons
6. Sixgrid : A minimal and battery friendly launcher for Android

Next steps.

- Be sure to include this cover letter in your GitHub repository
- Your GitHub repository should be tagged #AndroidDevChallenge
- Don’t forget to include other items in your GitHub repository to help us evaluate your submission; you can include prior projects you've worked on, sample code you've already built for this project, or anything else you think could be helpful in evaluating your concept and your ability to build it
- **The final step is to fill out this form to officially submit your proposal.**