## Tell us what your idea is

With phones constantly pushing the limits of the imaging capabilities of mobile cameras, recent years have seen a significant push towards mobile photography. As per [statista](https://www.statista.com/chart/10913/number-of-photos-taken-worldwide/), more than **1.2 trillion digital images** were captured in 2018, out of which, **over 85% were taken on a smartphone**. And while services like Google Photos can help people solve the storage problem, there is room for improvement when it comes to managing the clutter in each and every phone's gallery.

AfterShoot is a camera app that aims to be a second pair of eyes that analyses all the pictures in real-time as you take them.

It filters out the potentially bad pictures and recommends improvements that can be made to them.  
The recommendation engine has been trained on pictures captured by a variety of sources ranging from professionals to hobbyist photographers and various photography techniques like **Rule of Thirds, Rule of Depth** and the **Golden Ratio** to identify an aesthetically pleasing image from a bad one.

Aftershoot also packages a DeClutter module that can analyze and filter bad photos from your gallery which allows you to manage your storage efficiently.

Google's services I plan on using :

1. AutoML Vision Edge for training models to detect potentially bad and good pictures.
2. Tensorflow Lite to train a custom model that takes parameters for a bad picture and suggests potential fixes.
3. Firebase MLKit for hosting these models and running them locally.
4. Firebase for tracking app quality.

## Tell us how you plan on bringing it to life

I'm currently working on the models to identify and filter out good shots from the bad ones and so far I have:

1. Built the model that identifies blurry, overexposed, underexposed and pictures having considerable amount of people blinking : [**models/README.md**](https://github.com/harshithdwivedi/AfterShoot/blob/master/models/README.md)
2. Started working on the Camera and the DeClutter modules which will be integrated with the trained models above : [**aftershoot/README.md**](https://github.com/harshithdwivedi/AfterShoot/blob/master/aftershoot/README.md)

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

My 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**
  + Filter Duplicates – **Done**
  + Cropped Faces - **Done**
  + Sad Faces in an image, Foreign Objects in the picture - **Work in Progress**
* **January - February 2019:**
  + Add basic functionality to the DeClutter module – **Done**
  + Integrate the DeClutter module with the trained models - **Done**
  + 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.
  + Work on the Camera module
* **March 2019:**
  + Finish working on the camera module
  + Integrate the camera module with the trained ML models
* **April 2019:**  
  Bug fixes and model improvements
* **May 2019:**  
  Initial launch of the app

Most of the source code I currently have is for training the models above, so I won't be able to share it at the moment. Source code for the Camera and the DeClutter module, however, is open sourced and all the development done to the modules will be done in the open.

## Tell us about you

[Android Developer](https://harshithd.com/) and an [avid tech blogger](https://medium.com/@harshithdwivedi), I'm passionate about anything and everything related to Android. Currently a Google Developer Expert for Firebase; I am also [one of the first](https://www.youtube.com/watch?v=zfVoxusy-0M) Google-certified Android developers based out of 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.

Other than Google, I've been a regular contributor to Open Source Organizations like Apache, Redhat, and FOSSASIA. Recently I have been inclined towards my new found interest that combines my knowledge of Mobile Development with Machine Learning to create smart mobile applications.

Android Projects that I have worked on in the past:

1. [AiExhibit:](http://aiexhibit.org/) Worked with [HP Newquist](https://en.wikipedia.org/wiki/HP_Newquist) to build demos for America's first Artificial Intelligence Exhibition using Machine Learning to detect Objects, Predict Age, Identify Landmarks and Label Images.
2. [Offix-Android:](https://github.com/aerogear/offix-android) Worked as a maintainer of this open-source SDK that provided offline support and Conflict Resolution for people using GraphQL in their Android Apps.
3. [Coding Blocks Android:](https://github.com/coding-blocks/CBOnlineApp) Built an e-learning app which allows users to view, buy and download videos for the online coding courses offered by [Coding Blocks](https://codingblocks.com/).
4. [Open Event Android:](https://github.com/fossasia/open-event-droidgen) Worked on an Android App Generator project that helped Event Organizers to generate Android and Web apps for their event with a single click.
5. [Pokidex:](https://play.google.com/store/apps/details?id=app.harshit.pokedex&hl=en_IN) Android app that identifies Pokemon from the captured image, a real-life Pokedex.
6. [CrapCons:](https://play.google.com/store/apps/details?id=com.dagger.crapcons) An icon pack with a twist! CrapCons was the first icon pack for Android with all its assets made entirely in MS Paint.
7. [ZooperCrap](https://play.google.com/store/apps/details?id=com.adam.zwskin.zoopercrap&hl=en_IN): Widget extension for CrapCons.
8. [Sixgrid](https://play.google.com/store/apps/details?id=io.pure.sixgrid): A minimal and battery-friendly launcher for Android.