Please make a copy of this document and include this in your GitHub repository for your submission, using the tag #AndroidDevChallenge

Tell us what your idea is.

App Name: Live Parent
App main goal: making Android System safer and more accessible for Kids.
App in Detail:
The app **live Parent** is similar to the **live Caption** app but instead of using the audio stream, it will use the Video stream of the Android device that is being displayed on the screen.

This application should use ML to filter the content being displayed on the phone screen and categorize it in classes like G, PG-13, PG-17.
According to the class the application would respond:
G: Content will be allowed, no action.
PG-R or PG-17 … etc.: the application will blur the phone screen and:

- A warning message will be shown and ask for approval to show the content.
- The warning message could also ask for a password before unlocking the screen.
- If No is chosen the last active application or all applications would be killed or a list of active applications would be shown to kill the responsible application for the PG-R content.

The application would be very useful for parents who want to give their children more freedom using smartphones and surf the internet more freely but are afraid of their children being exposed to 18+ Content.
The application would be completely dependent on ML to determine if what is being displayed on the screen is not a +18 Content.
This Process, just like Live Captions, should happen in real time and completely on-device, so it works without cell data or Wi-Fi, and the pictures always stay private and never leave the phone.

Tell us how you plan on bringing it to life.

Describe where your project is, how you could use Google's help in the endeavor, and how you plan on using On-Device ML technology to bring the concept to life. The best submissions have a great idea combined with a concrete path of where you plan on going, which should include:

- *(1) any potential sample code you've already written:*
  - *I'm planning to use the open source app ScreenRecorder* https://gitlab.com/vijai/screenrecorder *as guidance to be able to capture the screen in the background.*
  - *Instead of saving it as video, I'll give it as an input to a MobileNetV2 which is trained to classify the pictures in (+3, +14, …) classes.*
  - *I'll use the TensorFLow image_classification example to build the Machine Learning part of the application and train it to classify the screen stream.* https://github.com/tensorflow/examples/tree/master/lite/examples/image_classification/android
- *(2) a list of the ways you could use Google's help:*
  - *I could use a lot of help from the team who made Live Captions, Live Parent uses the same concept of Live Captions but uses the video stream instead of the audio stream of the device. Getting their guidance and sharing their expertise with me would be very helpful.*
  - *Since my targeted audience segment is similar to the YouTube kids', the feedback of their team would be very essential to my app, especially because it's a very sensitive segment.*
  - *Google could provide me with a MobileNet V2 Model that is pre-trained on the +18 Content Dataset, or giving me access to the Dataset and the ML Model that google uses to automatically detect such content on YouTube/Google Play or other platforms.*
- *(3) as well as the timeline on how you plan on bringing it to life by May 1, 2020:*
  - *Start time 1. Dec 2019 analyze the ScreenRecorder open source app and image_classification example app.*
  - *Until 15 Dec 2019 the Live Parent app should be able to capture the screen in the background and save 1 frame every second on the memory.*
  - *Until 31 Dec 2019, import the pre-trained MobileNet V2 model from the TensorFlow image_classification example to the Live Parent app and use the saved frame on the memory as an input and classify it according to the dataset the model was trained on, in the case of the TensorFlow image_classification example the classes would be flowers.*
  - *Until 15 Jan 2020 using certain flower classes as place holder for +18 Content, which should trigger the blur/screen lock mechanism*

  - *Until 31 Jan 2020 alpha version of the app should be ready, and the functions of the Live Parent app, locking, unlocking blurring and unblurring the screen would be tested on flower place holder classes. The alpha version of the app should be ready by this time for HQ in Mountain View in case I get invited.*
  - *Until 29 Feb 2020 fine tuning a pre-trained Model on +18 Dataset and import it to the Live Parent App.*
  - *Until 15 March App modifying according to the Feedback of Google Dev Support*
  - *Until 15 April Testing the App.*
  - *Until 30 April Deployment of the App.*

Tell us about you.

My name is Alexander Monneret, I am at the last semester of studying bachelor of computer science in Albert-Ludwigs-Universität in Freiburg, Germany.
At the first Semester our group won the innovation prize in the System Design Project for the Robot that we built, since then Robotics and Machine Learning are my favorite subjects and I have never missed a chance to deepen my knowledge in those two subjects.

Experience in Machine Learning:

- Importing the End-to-end Driving via Conditional Imitation Learning experiment from Carla Simulator to AirSim Simulator and comparing the results, which included recording and preparing the dataset, building a neural network, train it and evaluate it performance.
- I have done the image classification and Object detection tutorials of TensorFlow Lite and tested the example app on my phone, including fine tuning and retraining on different datasets.

Experience in Android Programming:

- On 2017 I was awarded the Android Basics Nanodegree Scholarship from Udacity
- On 2018 I was awarded the Google Developer Challenge Scholarship from Udacity
- On 2019 TensorFLow Lite example apps and it's tutorials on my phone.