An ecosystem for evaluating Forensic Delay Analysis

KLINTERAI

Use Case: A Tensor Database for Construction Sector

**Describe the use case**

The Use case is described as per the document: “An ecosystem for measuring Forensic Delay Analysis.docx”

**Design and review the ML pipeline**

* **Use case requirements 1**: **Delay notification is sent to the Dashboard**

SRC: CCTV Video Images run through Visual Sentiment Analysis

C: IP Cameras, Relevant IoT Data through Messaging Queue installed in Cabin

M: Visual Sentiment Analysis Model installed in the Messaging Queue

D: Messaging Queue installed in Cabin connected to Cloud

SINK: The Dashboard which connects to IP Camera(s)

**SRC**: **Use case requirements 1**

For running Visual Sentiment Analysis, the Dataset is prepared using PyTorch CustomImageDataset. All the 1836 images are loaded onto Google Drive and the Google Drive is force mounted onto Google Colab for Practical Exercise purposes.

****

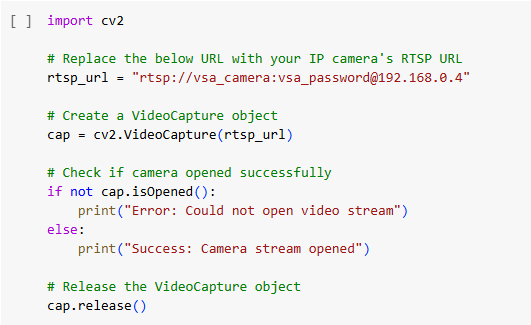
Train / Test Images are split into two annotations file(s): train\_anomaly\_detection.csv and test\_anomaly\_detection.csv. A batch size of 32 images is used to load the train / test set. For Image Classification, the classes (categories) used are:

* No Issue 0
* Risk 1
* Potential Risk 2
* Issue 3
* Potential Issue 4



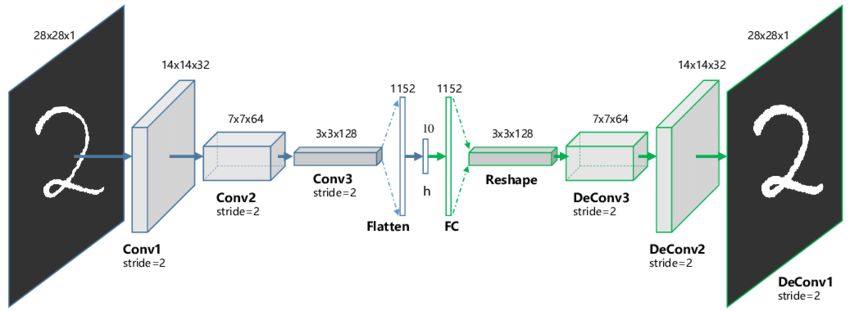
**C (Collector)**: **Use case requirements 1**

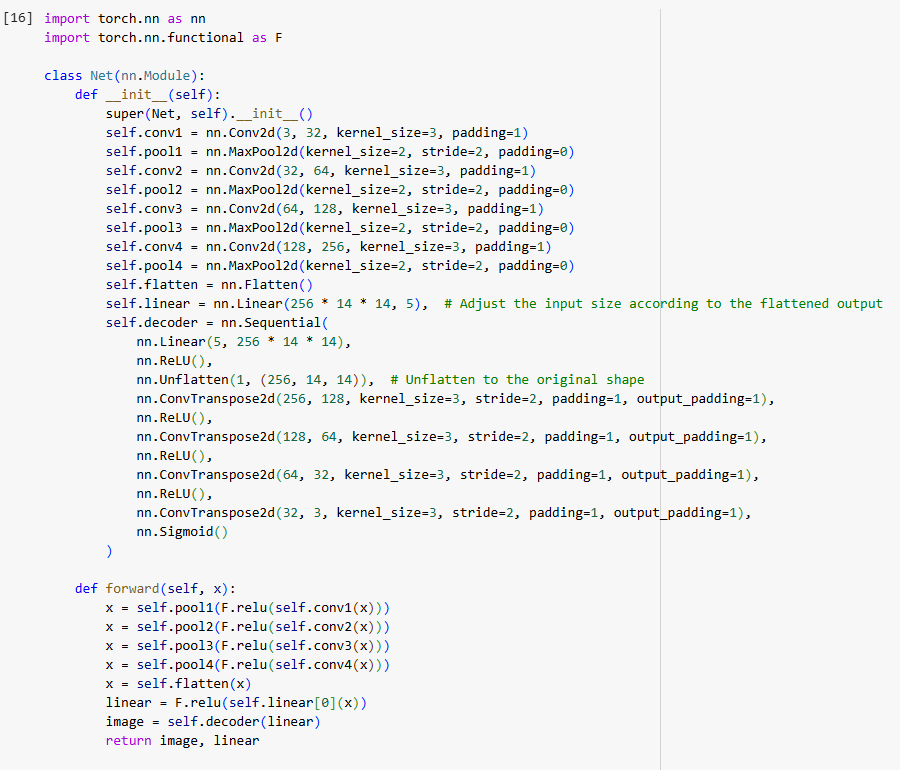
The MQTT Broker installed inside Cabin, with a Raspberry Pi-Like Device, connects to all the IP Cameras through RTSP (Real-Time Streaming Protocol). The MQTT Broker accepts other IoT Devices as well through Wireless technology. The Boilerplate code for accessing IP Cameras through OpenCV software is provided below:



**M (Model)**: **Use case requirements 1**

The Visual Sentiment Analysis (VSA) Model is installed inside the MQTT Broker and Messaging Queue operated under Raspberry Pi-Like Device. The Visual Sentiment Analysis (VSA) Model is an Autoencoder, a Convolutional Autoencoder with 5 features in the hidden layer specifically designed for image classification.

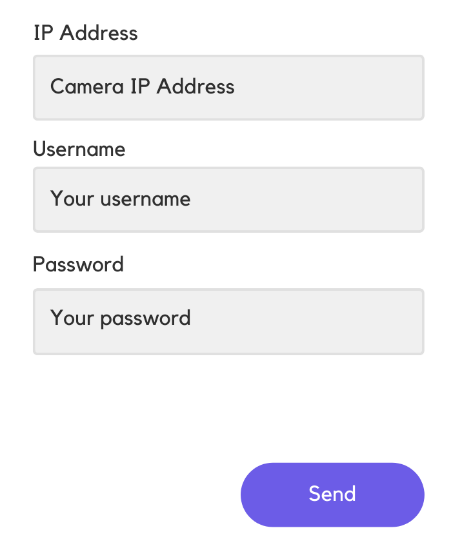




The Model is designed using PyTorch Autoencoder and integrated into the Raspberry Pi-Like Device used as Messaging Queue, MQTT Broker and OpenCV Inference Box from the feed of IP Cameras.

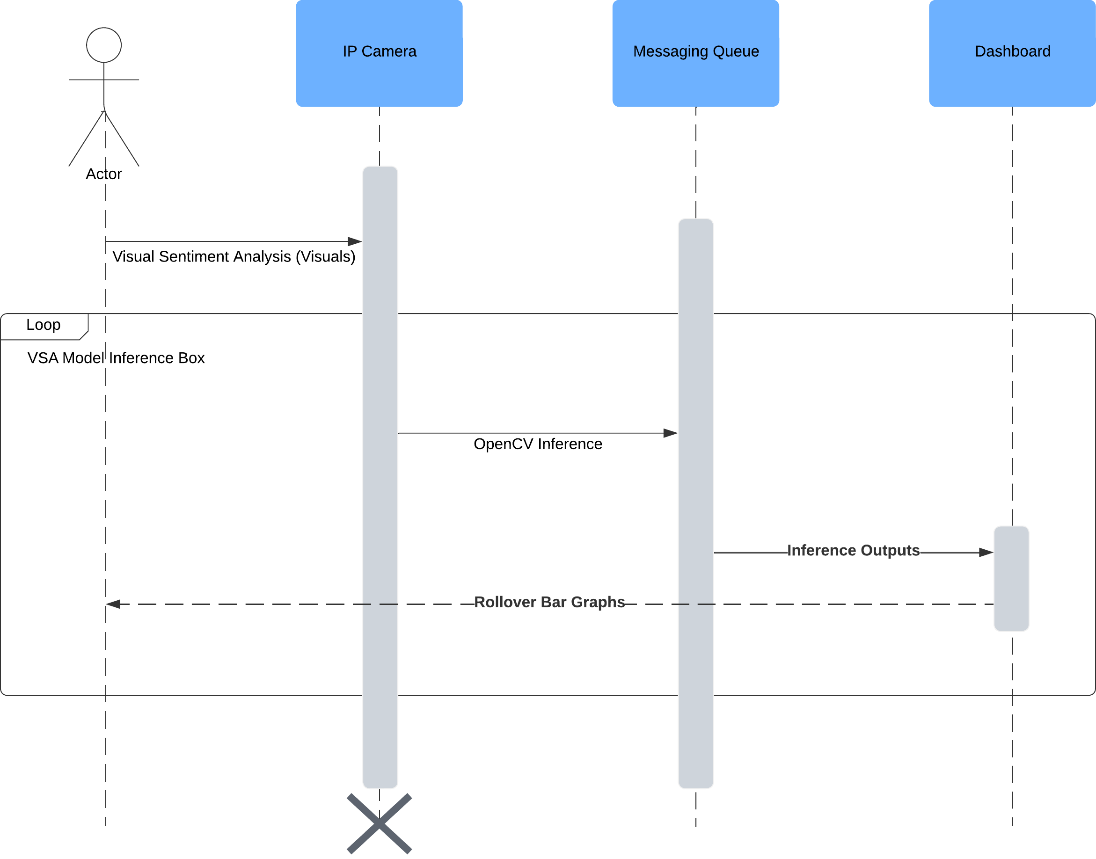
**D (Distributor)**: **Use case requirements 1**

The Raspberry Pi-Like Device is the Distributor here, sending information aggregated from the IP Cameras to the Cloud Dashboard. The Cloud is connected to the Distributor in a 2-way linking. The RTSP Protocol is instantiated inside the OpenCV Inference Box or the Device using a web form attached to the Cloud Dashboard that send information to the Device attached to the Construction Site Cabin. A Cabin is mandatory for all projects in Construction Industry as per Health & Safety Standards, hence this Inference Box is part of the Cabin.



**SINK (Dashboard)**: **Use case requirements 1**

The Dashboard is the xApp from ITU.3061 Standards. The Dashboard connects to the Messaging Queue using a Cloud Database readable from the Messaging Queue.



**Design and review the ML pipeline**

* **Use case requirements 2**: **Site Engineer addresses the Risk via Delay Notification Received at Mobile Device**

SRC: CCTV Video Images run through Visual Sentiment Analysis

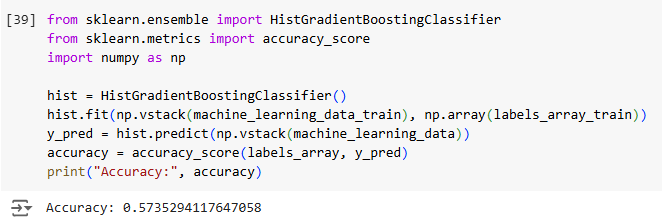
M: Visual Sentiment Analysis Model installed in Cameras or the Messaging Queue

P: A Proper Rollover Metric Policy attached to the Model that signifies a Risk from the Delay in the Presence of an Activity

D: Messaging Queue installed in Cabin

SINK: The EquipAny Mobile App

**P (Policy)**: **Use case requirements 2**

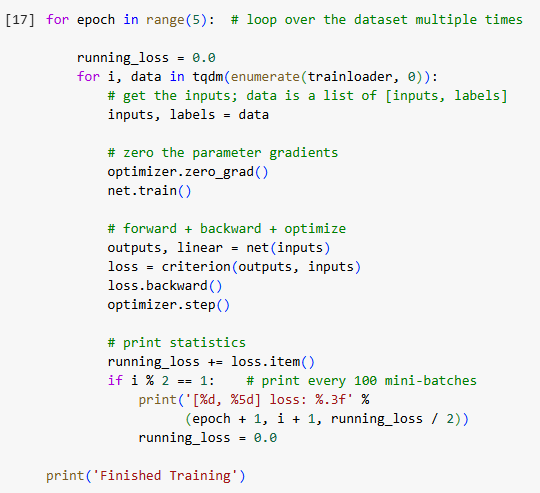


The output of the Model node in the ML Pipeline is subjected to Risk analysis. Half of the outputs correspond to Risk and Issue and the other half correspond to Potential Risk and Potential Issue. In-effect this is an Anomaly Detection task, which will fetch even more higher accuracy than image classification or sentiment analysis task. The right Policy is to apply for a Rollover metric to reduce the number of frames coming out of the OpenCV Inference Box Video Frames. The Rollover Metric that must be used is Effective Rollover Sum to perform that reduction.

A graph of different colored bars

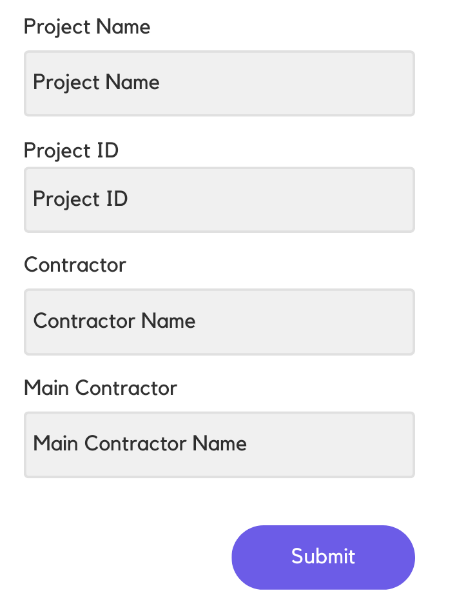
Description automatically generated with medium confidence

The training loop of the ML Pipeline is provided below. It uses a loss of MSELoss (Mean Squared Error Loss) and uses the feature representation layer to train with labels and produce a minimum accuracy of 0.57.



**SINK (EquipAny)**: **Use case requirements 2**

The EquipAny mobile app is the rApp of the ITU.3061 Standards. EquipAny is used on demand and does not listen to the Real-Time output or warnings of the Visual Sentiment Analysis (VSA) Model. The EquipAny Mobile App Screen is shown below: to reflect on identifying a Generic Issue as per the Screenshot. EquipAny Mobile App asks the Project Name, Project ID, Contractor, and Main Contractor as the initial set of attributes.





**Build the ML Pipeline**

The ML Pipeline is built at Github Repository: KlinterAI with link:

<https://github.com/KlinterAI/klinterai-website>

**Demo**

Please refer to the Code Walk through Video for the Demo in this section.

<https://Klinterai-tenant.vercel.app>

<https://Klinterai.carrd.co>

The above two sites are KlinterAI showcase site and crowdfunding site respectively.