# **LICENSE PLATE DETECTION**

# 

ABSTRACT:

The purpose of this project is to build a data pipeline that programmatically downloads the most up-to-date images of license plates, using the Google API for Images. Once we fetch the images, we pass them to our API service which provides us with the predicted value of the text in the images. These predicted values are validated using Amazon’s MTurk service in the inference part of this pipeline.

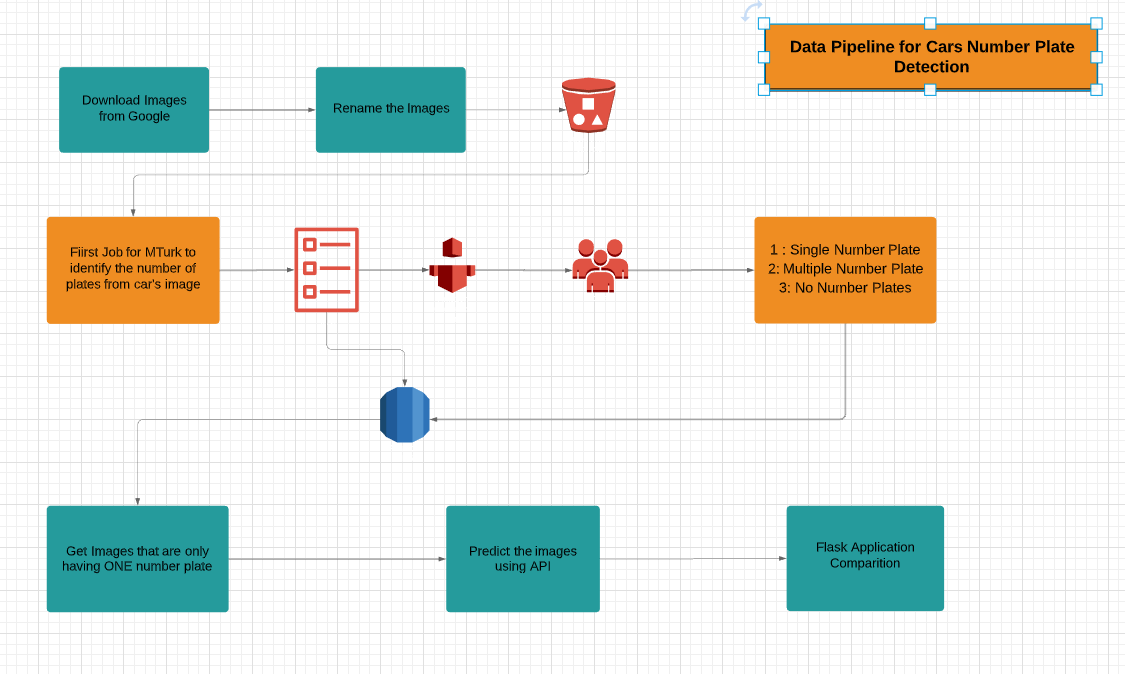
# **PROJECT GOALS**

* Programmatically download images from the Google API
* Rename and store the images in Amazon S3 Bucket to create a data lake
* Send all images to MTurk for classification into Single, Multiple and No plates
* Select only images with single license plates and pass it programmatically to our model API
* Validate our predicted values using MTurk Inference pipeline
* Display side-by-side results to user on a Flask Front End

# 

# 

# **DATA PIPELINE**



## STEP - 1 : Data Scraping

* The data set was programmatically downloaded from Google using its Python API
* Renamed and stored the images in Amazon S3 landing Bucket to create a data lake
* This pipeline is scheduled using Apache Airflow to run every 10 mins to give us 50 new images

## STEP - 2 : Data Validation using MTurk

* MTurk create HITs for tasks to classify and label images and store HITs metadata in RDS
* We fetch the responses for each image from 2 distinct workers so that there is lesser probability of user error and store the responses of all the images as tags to the images in the RDS instance

## STEP - 3 : Prediction using Model API

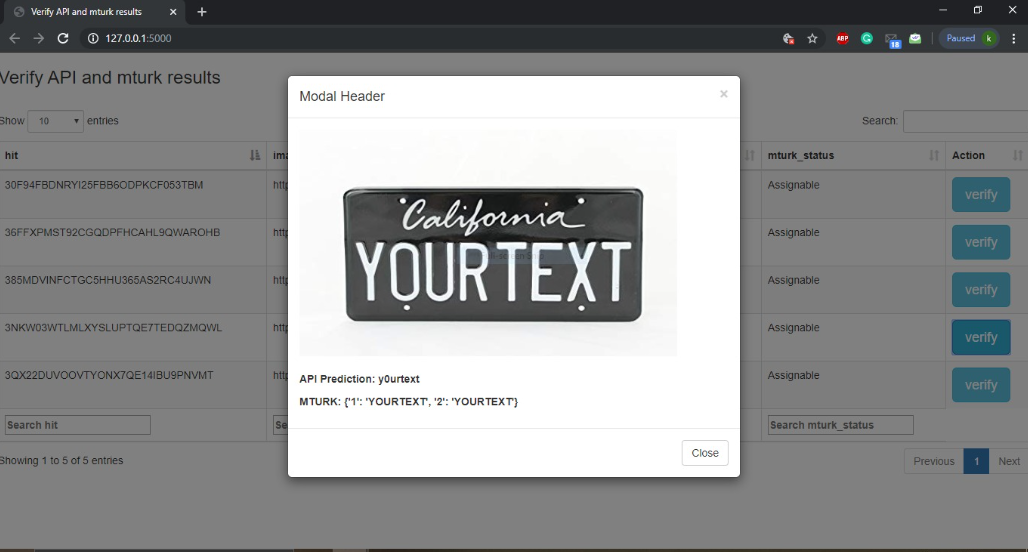
* We select only those images where there are only single license platesin the images
* Next, we programmatically send them to our API for detecting the predicted text values in the license plate

## STEP - 4 : Inference Validation using MTurk

* Once the predictions are received from our API, we validate the predictions by comparing these values from those that we have received via the MTurk workers

## STEP - 5 : Flask Front End

* After receiving the predictions from the MTurk and the API, we display the side-by-side comparison on a user-friendly Flask Front End



# 

# 

# REFERENCES

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/mturk.html>

<https://flask.palletsprojects.com/en/1.1.x/quickstart/>

<https://datatables.net/manual/>

<https://www.openalpr.com/cloud-api.html>