

Assignment # 3: AutoML (no-code/low-code Machine Learning)
Course: CAP 6610 Applied Machine Learning
Professor: Dr. Oge Marques
Student Number: Z23596812
Student Name: Renee Raven
Term: Fall 2022

Report: Part 1

Tool 1 : Lobe

Lobe is a friendly site that incorporates bite size info in big font and nice graphics. It targets someone new to ML. It offers image classification (label an image based on its content). Future development includes object detection (locate an object inside of an image) or data classification (label data in a table based on its content). A highlighted testimonial uses Lobe for improving models of nature-dependent tourism. One user states it is easy enough for his kids to use and that the kids enjoy it. Additionally, Lobe offers several ways to export the app once trained.



The “Introducing Lobe” video promises: “Build your first machine learning model in ten minutes.” I followed the instructions to create a machine learning model to recognize drinking after downloading the app to my mac laptop.

1. Add the name of the task “Drink Tracker”
2. Go to import and use the camera function on the laptop to take pictures while drinking from a glass
3. Label those images “Drinking”
4. Take pictures while not drinking, perhaps mimicking drinking but without a cup
5. Label those images of “Not drinking”

I tested the project with poor lighting. The program started training automatically. Then I switched to the use function where a live picture from the camera is deemed “Drinking” or “Not drinking.” Initially, it made several mistakes, but they were easy to correct with the red slash button or green check mark button.

Drink Tracker

 Label

 Training... 

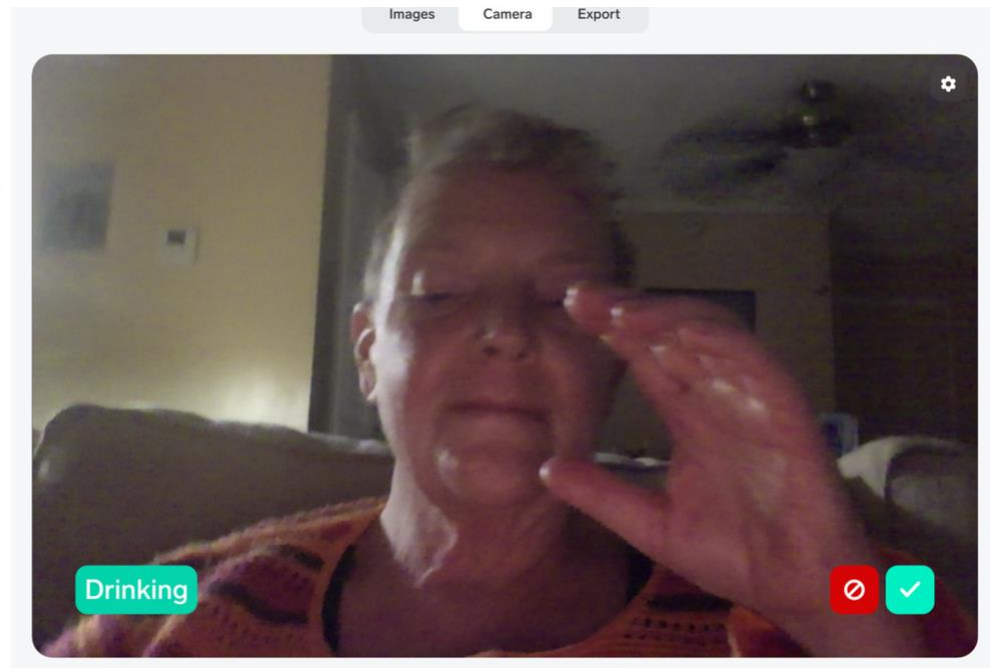
 Use

All Images 93%

Drinking 98%

Not drinking 87%

85% of your images are
predicted correctly,
15% incorrectly.



With each correction, the model retrained and the performance greatly improved. Eventually, it correctly predicted 99% of images.

Drink Tracker

 Label

☒ Train

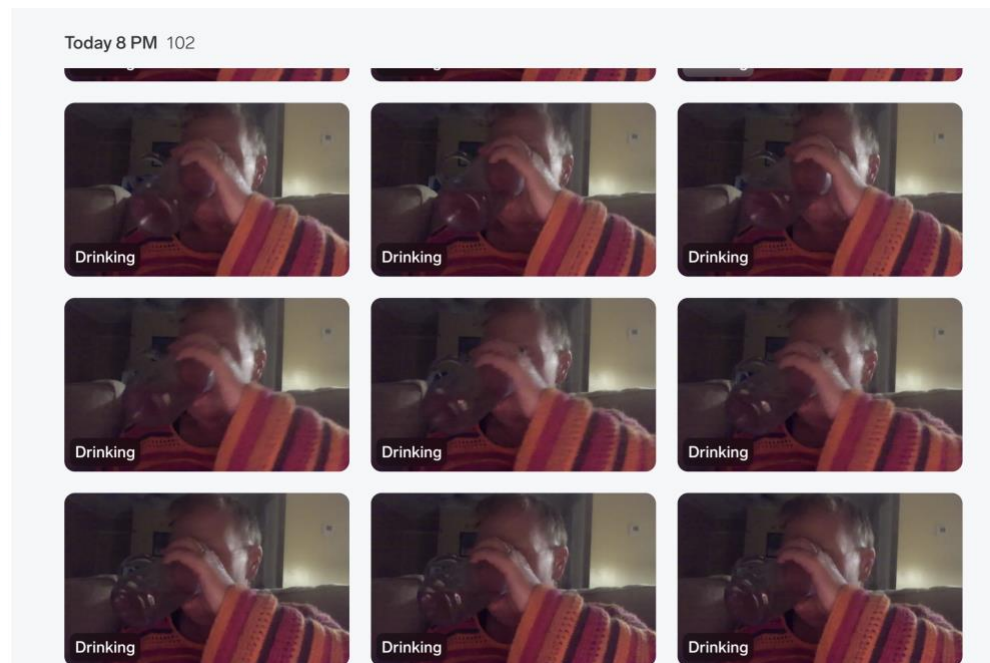
 Use

All Images 203

Drinking 102

Not drinking 101

99% of your images are
predicted correctly,
1% incorrectly.



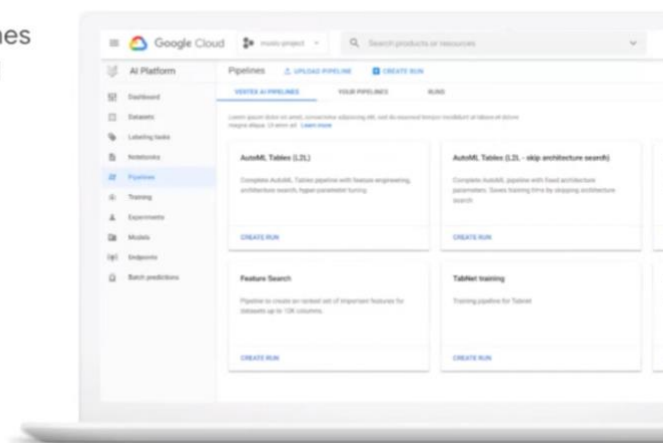
Tool 2 : Google AutoML

Google AutoML targets software developers with some understanding of machine learning, but not enough to utilize it for business purposes. AutoML helps users create their own custom, high-quality, models to answer their business concerns. The site offers a short video which introduces tabular workflows which are scalable pipelines that minimizes complexity and high maintenance costs. Endorsements include Lowe's (they use it for inventory tracking) and General Mills (usability, scalability, and reliability).

Introducing Tabular Workflows

Integrated, fully managed, scalable pipelines and individual components for end to end model development with Tabular Data.

- AutoML
- Managed Algorithms
- Feature Search
- Feature Transformations



The video also talks about the 2 most common problems with standardized ML pipelines: lack of transparency and lack of flexibility. They suggest AutoML workflows provide flexibility (modifiable) and transparency (detailed information in logs and artifacts).

AutoML workflows are accelerating ML

Automated

AutoML automates repetitive and manual machine learning tasks. In many cases (like hyperparameter optimization) **AutoML can save development time and improve model performance.**

Flexible

Workflows can be modular and allow high degree of customization. You can **skip, modify specific steps of the workflow and change hardware options.**

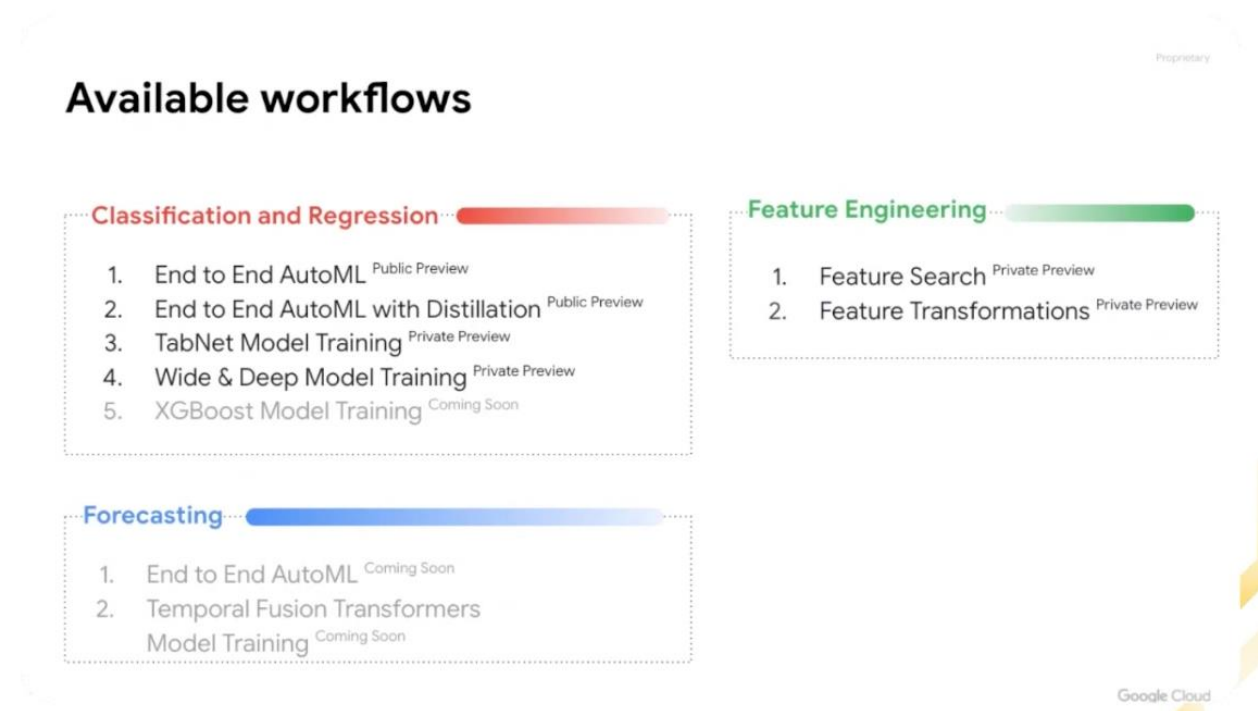
Transparent

Workflow pipelines are transparent. You can easily **see detailed information about each of the steps**, the logs from the triggered services and **the artifacts created** by each of the workflow components.

Integrated

Workflows **integrate with MLOps tools** like Feature Store, Model Metadata and Experiment Tracking. Increase the number of experiments per day without being weighted by the management and monitoring overhead.




Google AutoML offers a menu of classification, regression, and feature engineering tools. Forecasting, including end to end AutoML and temporal fusion transformers model training, are in development.



Tool 3 : Google ML Kit +

Google ML Kit is aimed at the mobile app developer. It brings Google's "on-device machine learning expertise to Android and IOS apps." It offers face detection, text detection and object detection integration, as evidenced by the highlighted case studies:

Case studies

		
ZYL	LOSE IT!	ADIDAS
Learn how Zyl used ML Kit's face detection and image labeling APIs to shrink their app by 50%.	Read about how Lose It! used ML Kit's text recognition API to improve the user experience.	Learn how adidas uses ML Kit's Object Detection & tracking API to enable visual search in their stores.
More info	More info	More info

ML Kit offers sample apps that are either quickstart (simple examples to launch quickly) or showcase apps (combines multiple ML Kit APIs with Material Design components to generate polished apps), all optimized for mobile.

Two of the more interesting sample apps are pose detection (real time identification of the body position) and face mesh detection with 468 3D points and edges. Use cases of face mesh include real-time AR filters, selfie capture, and video chat.



Face mesh detection **NEW**

Detect face mesh info on close-range images.

[Get started](#)


But Google ML Kit offers a wide range of capabilities including: face recognition, text recognition, translation, image labelling, object identification and tracking, barcode scanning, and smart reply where replies are automatically generated based on the flow of text conversations.

Report: Part 2

I was pretty impressed with the simplicity of Lobe.ai, so I looked forward to challenging it with images from 2 categories that are fairly close. Indoor scene recognition from MIT offers a collection of 67 image categories where each category contains at least 100 unique images.

<http://web.mit.edu/torralba/www/indoor.html>

← → ↻ 🏠 🔒 web.mit.edu/torralba/www/indoor.html 📄 ☆ 🔍 Search



Store	Home	Public spaces	Leisure	Working place
bakery, grocery store, clothing store	bedroom, nursery, closet, pantry	prison cell, library, cloister, church	buffet, fastfood, concert hall	hospital room, kinder garden, restaurant, kitchen, artstudio
deli, laundromat, jewellery shop	children room, lobby, dining room, corridor	waiting rooms, museum, elevator	restaurant, bar, movie theater	classroom, laboratory, workshop, studio, music, operating room
bookstore, video store, florist	livingroom, bathroom, kitchen	pool inside, inside bus, inside subway	casino, bowling	office, computer room, warehouse, green house
shoe shop, mall, toy store	staircase, winecellar, garage	locker room, trainstation, airport inside	gym, hair salon	dental office, tv studio, meeting room

Indoor Scene Recognition

Indoor scene recognition is a challenging open problem in high level vision. Most scene recognition models that work well for outdoor scenes perform poorly in the indoor domain. The main difficulty is that while some indoor scenes (e.g. corridors) can be well characterized by global spatial properties, others (e.g., bookstores) are better characterized by the objects they contain. More generally, to address the indoor scenes recognition problem we need a model that can exploit local and global discriminative information.

Database

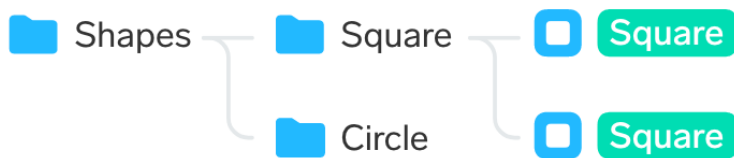
The database contains 67 Indoor categories, and a total of 15620 images. The number of images varies across categories, but there are at least 100 images per category. All images are in jpg format. The images provided here are for research purposes only.

[Download \(tar file, 2.4 Gbytes\)](#)

This particular dataset uses labelled folders to collect the images, making it ideal for working with Lobe.

Label Images

Would you like Lobe to use the folder names to automatically label your images?

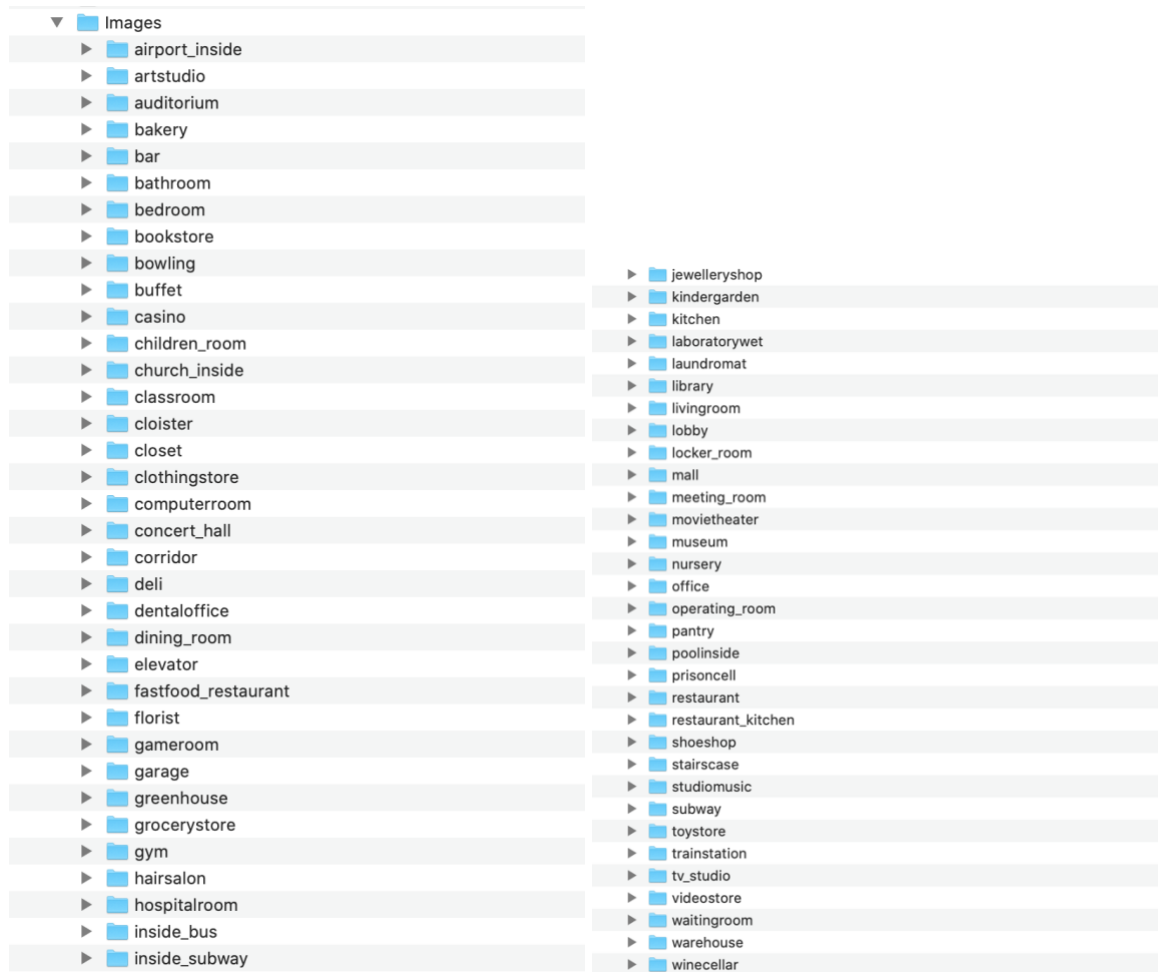


- ☒ **Label Using Folder Name**
Use folder names to label your images.
- ☐ **Label Manually**
Use Lobe to manually label your images.

Cancel

Import

From the 67 potential categories, I selected indoor_bus and indoor_subway, since I suspected they would have similar images.



I then extracted the first 5 from each folder to perform a manual test after the model was trained. Lobe rejected 2 of the images. In total, 97 of the 98 available images from inside_bus folder were uploaded and 452 of the 453 available images in inside_subway folder were uploaded.

subway_bus

Label

☒ Training...

☐ Use

All Images 549

inside_bus 97

inside_subway 452

Lobe is starting to train your model...

All Images

View

Import

Show Less

inside_bus 97



Training took less than 10 minutes and progress was updated periodically.

subway_bus

Label

☒ Training...

☐ Use

All Images 549

inside_bus 97

inside_subway 452

18% of your images are predicted correctly,
82% incorrectly.

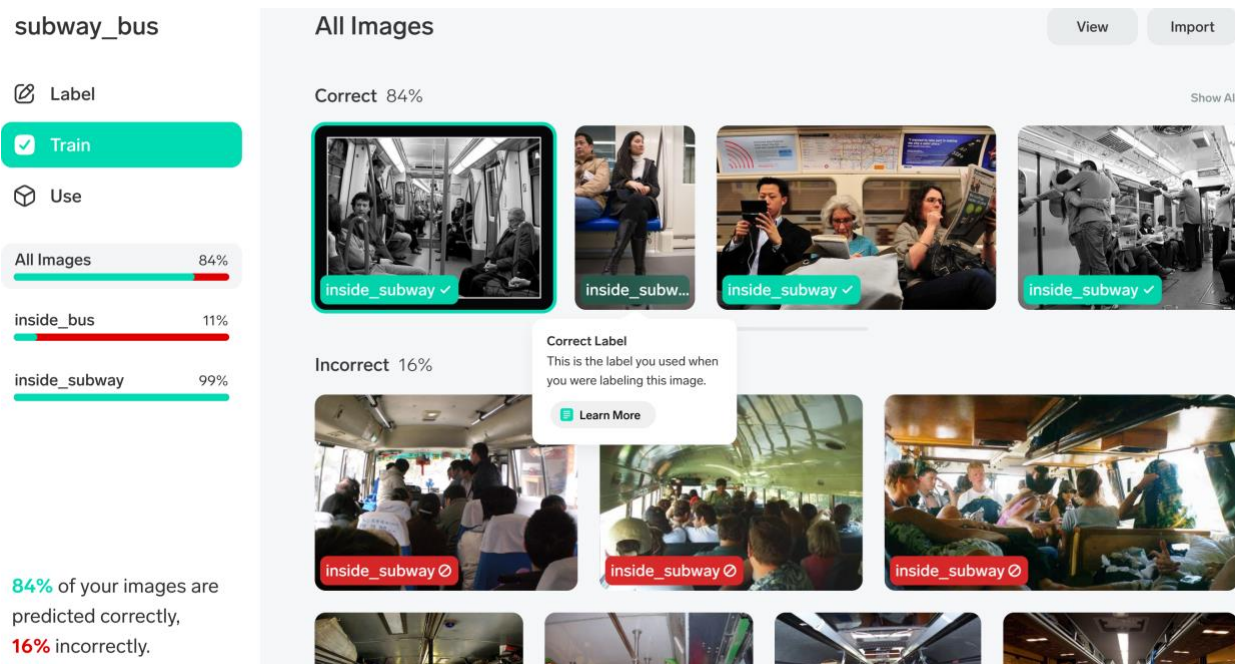
All Images

View

Import

inside_bus 97

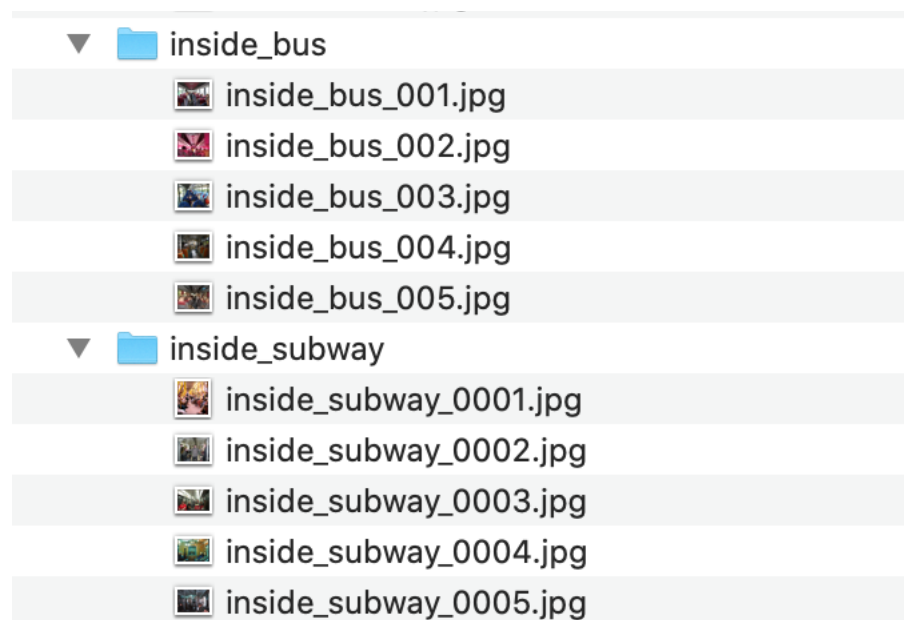




84% might sound like a great number, but Lobe didn't actually perform so well on identifying the inside_bus images (only 11%). This imbalance is likely due to the imbalance in the size of the datasets (97 vs 452 images).

This is a real weakness of the model. Unfortunately, someone who is naïve to ML problems might not catch why this is so imbalanced and think they have created a pretty model.

Then I tested the model with the 10 sample images.



The Use function was used to check the test images. The first image from the inside_bus test files was loaded. Every time a new image was loaded, the user was prompted to indicate if it was right (red slash circle) or right (green check mark).

subway_bus

 Label

 Train

 Use

All Images 84%

inside_bus 11%

inside_subway 99%

84% of your images are
predicted correctly,
16% incorrectly.



2/5 of the inside_bus images were incorrectly identified. 3/5 of the inside_subway images were incorrectly identified. During testing, the percentage of correct/incorrect flip-flopped. It seemed to depend on whether the model was over predicting inside_bus or inside_subway.

subway_bus

 Label

 Train

 Use

All Images 28%

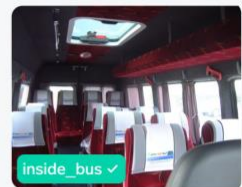
inside_bus 98%

inside_subway 12%

28% of your images are
predicted correctly,
72% incorrectly.

Incorrect 72%

Correct 28%



subway_bus

Label

Train

Use

All Images 91%

inside_bus 55%

inside_subway 99%

91% of your images are predicted correctly, 9% incorrectly.

Incorrect 9%

Correct 91%



And the final percentage, after all 10 test images were loaded, was pretty dismal.

subway_bus

Label

Train

Use

All Images 18%

inside_bus 100%

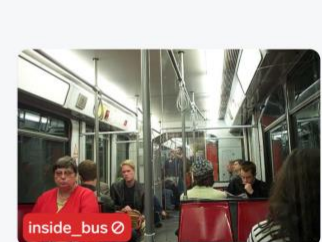
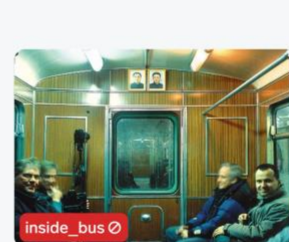
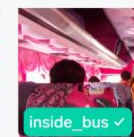
inside_subway 0%

18% of your images are predicted correctly, 82% incorrectly.

All Images

View

Import



With a total of 5/10 test images correctly identified during the testing phase, it is clear this model performs at chance level and has not developed an algorithm to make useful predictions. This may be because of the imbalanced datasets. It would be interesting to see if it did better with 2 categories with an equal number of images and/or 2 categories that are more dissimilar during future experiments.