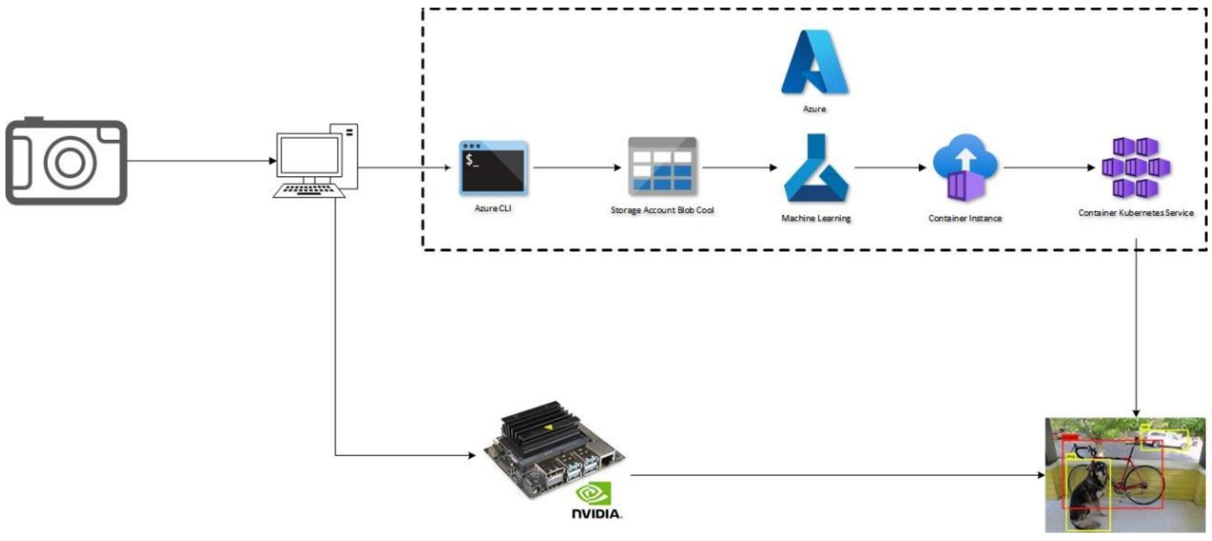
Exercise 2: Class Auto labelling Approach

Question: You have access to a large-scale image dataset with no labels or meta data, we need a solution to automatically detect class labels for these images and build a corpus of classes. Define your solution and approach and indicate a sample pipeline to achieve this.

Answer:

End to End Flow: Camera that captures Images and Saves in Local system.

- Push to Cloud storage or Jetson, based on Requirements



Once Data is been stored we can Start building our Model

Solution 1: (General Objects)

If image Dataset includes General Objects like People, Vehicle, Bag, Laptop, cup etc

We can straight way use - COCO - Common Objects in Context (cocodataset.org) Pretrained model which can detect 80 general classes and customized to only classes we need.

Pretrained Model for TF 2.X : models/tf2_detection_zoo.md at master · tensorflow/models · GitHub

- Suitable for Object Detection (Boxes), Key points, Segmentation.
- We can further write script to save the Coordinates(x,y,h,w) or save images in specific folders (OS, Glob) based on class detected.

Hence of any General object to detect Class labels, we can using Pretrained Models which are already available and build a corpus of classes of all images

Cons:

- However, for Domain Specific objects like Vehicle parts, Chipset or Electrical components usually Robust pretrained models are not available.

Domain specific images and Objects classes

Solution 2: Semi Supervisied Learning

Eg: lets assume we have access to 1lakh images of different parts.

- Build a Based model for Image classification by using 5k or maybe 10K images labelled Manually by Human.
- Its always to good to use Transfer Learning apporach in such cases for quick model training and high accuracy.

Below is a link for using pretrained models (resnet,inception,mobilenet and 20+ models)

Module: tf.keras.applications | TensorFlow Core v2.9.1

E.g., usage of InceptionV3:

```
tf.keras.applications.inception_v3.InceptionV3(
    include_top=True,
    weights='imagenet',
    input_tensor=None,
    input_shape=None,
    pooling=None,
    classes=1000,
    classifier_activation='softmax'
)
```

- And Finally you can train and fine tune your model in tensorflow using below pipeline Transfer learning and fine-tuning | TensorFlow Core

- Aim is to get accuracy of 85+ or 90+ for Base model by trying different Hyperparameter, data Augmentation etc. and after getting good accuracy you can Run your base model on Remaining 90K models.
- Now 90K images which be classified and labelled and now you can train another model with 1Lakh labelled data (10K manually labelled and 90K Labelled by Baseline Model.

Above Approach is more robust and highly recommended because the base model is trained on (10K) data labelled by humans or Domain specific experts.

If we are able to achieve High accuracy in Base model, High chances to classify test set or unlabelled data (90K) accurately.

Cons:

- However, this approach still requires to Manually label few samples data by humans to built base model.

Solution 3 : UnSupervisied Learning

I. Using Feature Extraction technique and then Clustering the feature maps.

Extract Features using some Base models

```
model = VGG16()
model = Model(inputs = model.inputs, outputs = model.layers[-2].output)

def extract_features(file, model):
   # load the image as a 224x224 array
   img = load_img(file, target_size=(224,224))
   # convert from 'PIL.Image.Image' to numpy array
   img = np.array(img)
   # reshape the data for the model reshape(num_of_samples, dim 1, dim 2, channels)
   reshaped_img = img.reshape(1,224,224,3)
   # prepare image for model
   imgx = preprocess_input(reshaped_img)
   # get the feature vector
   features = model.predict(imgx, use_multiprocessing=True)
   return features
```

Once Features are extracted Run a clustering algorithm on the Feature Matrix using Clustering Algorithm like K-means.

II. Semantic Similarity of the Images

E.g.: Finding similar images using Deep learning and Locality Sensitive Hashing | by Aayush Agrawal | Towards Data Science

Have reference class images and check for similarity of the Images :

E.g. : How to Implement Image Similarity Using Deep Learning | by Sascha Heyer | Towards Data Science