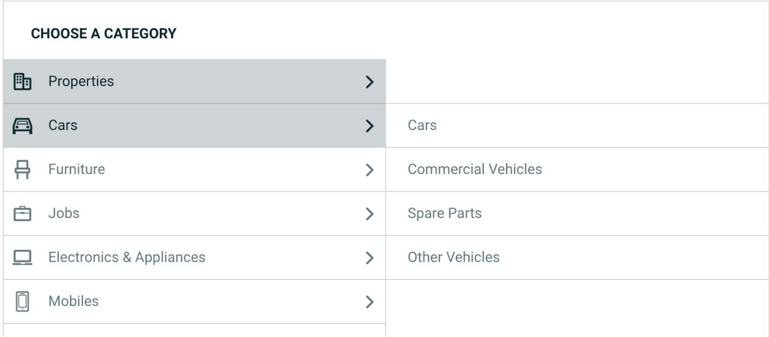
Domain Background

In OLX we build leading destinations for buying, selling, and exchanging products and services, spanning across 5 continents and used by 350 million people every month.

Currently, if a customer in OLX wants to post a product, they have to select a category first. Selection of categories is where the major portion of time customers spend. We did some research on why part and found some interesting pointers raised by customers.

- *I don't understand what you mean by categories*
- *As you have 40 odd categories to select from, it is a hard time to find the relevant category*
- *My definition of categories is not aligning with your definition of categories*



This is where we as OLX see the opportunity to use ML to solve the customer problem of selecting the product category automatically. Through the image classification algorithm, OLX wants to read the picture that the seller inputs to automatically classify the product into its correct category. Having the correct category for the product will make the experience easier for OLX customers to post.

Problem Statement

- **Problem**
  I Want to build a good image classifier that will help in automatically classifying the correct category for the given image of the product.
- **Solution**
  For image classification algorithms, I will be using Convolutional Neural Network(CNN), because of it's a great performance. The algorithm will be trained with a large amount of high-quality image data to get good results

I will be taking 100K images from OLX equally distributed among categories which I will use to train my model and should be able to classify products correctly into one of 5 categories.

Just for clarification purpose, OLX has 40+ categories, but I am targeting top 5 categories because of 2 main reason

- Fetching and storing images for all categories to create dataset will be of huge task and ROI will not be that high.
- By picking the top 5 categories, I am almost catering to 90% of our requests.

The goal of the system is to achieve high accuracy in terms of predicting the correct category, and if accuracy is low, at least I can limit the customer to relevant categories whose accuracy score > 50 % and < 90%

Datasets and Inputs

I will use the dataset from OLX, a new image dataset for benchmarking ML algorithms. The dataset consists of a training set of 80K and a test set of 20K.
Each dataset will contain an image with an associated category/class. The dataset has 5 categories/classes: Car, Mobile, bikes, books and furniture.
Dataset will be equally distributed across categories, each category will have 20K images associated with it.
Sample image of the bike category

Solution Statement

CNN is best in categorizing image data. CNN also helps you in combining layers to create complex non-linear classifiers. Basically to identify/distinguish unique patterns within images.

I will be relying on transfer learning to build an accurate model. By transfer learning, I can start from patterns that have been learned when solving a different problem. This way I will leverage from past learning and customize according to my needs by retraining it.
I will be using Amazon Sagemaker in-build algorithms for CNN (ResNet) for image classification which will be trained using transfer learning. It won the 2015 ImageNet Large Scale Visual Recognition Challenge for best object classifier.

Benchmark Model

As this is a classification problem where upon given a test dataset, I have to classify it to one of the 5 categories/classes. For that, I will use the Convolution Neural Network(CNN). There are ways to create CNN models, but for benchmark, I will be using Keras deep learning library pre-trained model **ResNet50** and **InceptionResNetV2** for my task.
I will benchmark the dataset with a simple and high-end model to prevent overfitting/underfitting the dataset on a given model.

Evaluation Metrics

Metric that I will be using to quantify the performance of both the benchmark model and solution model proposed.
Will be using the accuracy metric. This is a common evaluation metric in classification problems and it is used since data is distributed equally among classes/categories.

Project Design

This is how I will do the design part

Prepare my dataset

To get the best results, data needs to be balanced across categories/classes.
To make sure the above statement stands true, I will download 100K images from OLX, which have an equal ratio among categories.
The Amazon SageMaker built-in Image Classification algorithm requires that the dataset be formatted in RecordIO. First I will convert the raw images to RecordIO, and after that, I will upload them to S3.
To prepare the dataset:

1. Unpack the images to raw JPEG grayscale images of 28×28 pixels.
2. Organize the images into 5 distinct directories, one per category.
3. Create two .lst files using a RecordIO tool (im2rec). One file is for the training portion of the dataset (80%). The other is for testing (20%).

4. Generate both .rec files from the .lst
5. Copy both .rec files to an Amazon S3 bucket.

Set up the environment

Will be following up the same way I created an environment for Plagiarism detector and Sentiment analysis in AWS. Creating ROLE attached to the notebook which has access over S3 How my trained model behaves will be strongly affected by the way I set hyperparameters. The parameters that I will be playing with is

1. Epocs
2. Transfer Learning

I have to select EC2 with a higher GPU to train my model.

Training model

To train, first, I will try to select the image the Sagemaker provides to run image classifier. I will use that image to train my model and to publish the model in production as an endpoint.
Then I will try to run Sagemaker job with all parameters and hyperparameters set and save the model in S3.

Validating the model

Then I will be validating my model with the benchmark I have for accuracy.

Links

https://arxiv.org/abs/1512.03385
https://docs.aws.amazon.com/sagemaker/latest/dg/image-classification.html
https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/
https://github.com/bonn0062/image_classifier_pytorch
https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90
https://docs.aws.amazon.com/sagemaker/latest/dg/image-classification.html