Capstone Proposal

Deepak Batra

20th June 2018

Proposal

Domain Background

Myntra is India's largest fashion e-commerce company. A variety of products are listed on myntra.com/web/Myntra's mobile application and bought by the customers using the platforms. A customer purchases a wide variety of clothing, be it shirts, t-shirts, jeans etc. Product (clothing) attributes influences a customer's decision process; be it checks on the shirts, patterns on the jeans or graphics on the t-shirts. To make sure that the right inventory is available and the right products are personalised as per the customer's intent/ buying history Myntra needs to extract attributes from the clothing's images and divide products into categories which can then be linked to customers and therefore the content can be personalised per customer. Also, this would help in inventory management as the products which are high on customer intent will be bought more and therefore increasing saleability on the platform.

Problem Statement

As discussed above, deciphering attributes from an image of clothing is a problem that can help the business in many ways than one. But categorising all types of clothing requires a big train/test dataset and therefore a smaller **classification problem** is chosen which is to identify and categorise the graphic type of a **t-shirt** type clothing only. However, the same can be extended to other types of clothings be it **shirts**, **jeans** etc.

To define the problem, a **t-shirt** is going to be categorised into 24 different pre-defined categories. The categorising model will be used in cataloguing new t-shirts in Myntra by extracting graphic-type attribute and therefore a customer can search/filter based on this attribute from the mobile app/web client. The model takes a set of images and predicts its category-type. The datasets are divided into **test** and **train** data and the **accuracy metric** of the data is going to be the **percentage prediction** in images in the test-data.

Datasets and Inputs

The data is divided into two datasets namely **test** and **train** dataset. The features being used are as follows:

`Brand, Article-type, Gender, Colour, Image, [Graphic Type`[1]`]`

Brand is an integral part of the features as specific brands are known for their articles/ graphic-types.

Article-type (ex. t-shirt/shirt etc) is included so as to extend this model to predict the categories for shirts/jeans later on.

Gender is a again a determining factor in identifying the graphic of type

Image from which the graphic-type is extracted and category is found.

Solution Statement

A possible solution to the problem is to use **ResNet** or **VGG-16** and to add a 2-3 dense layers at the end, therefore training the network on the **train-dataset** and evaluation to be done on the **test-dataset**. A possible network architecture might be similar to the one given below:

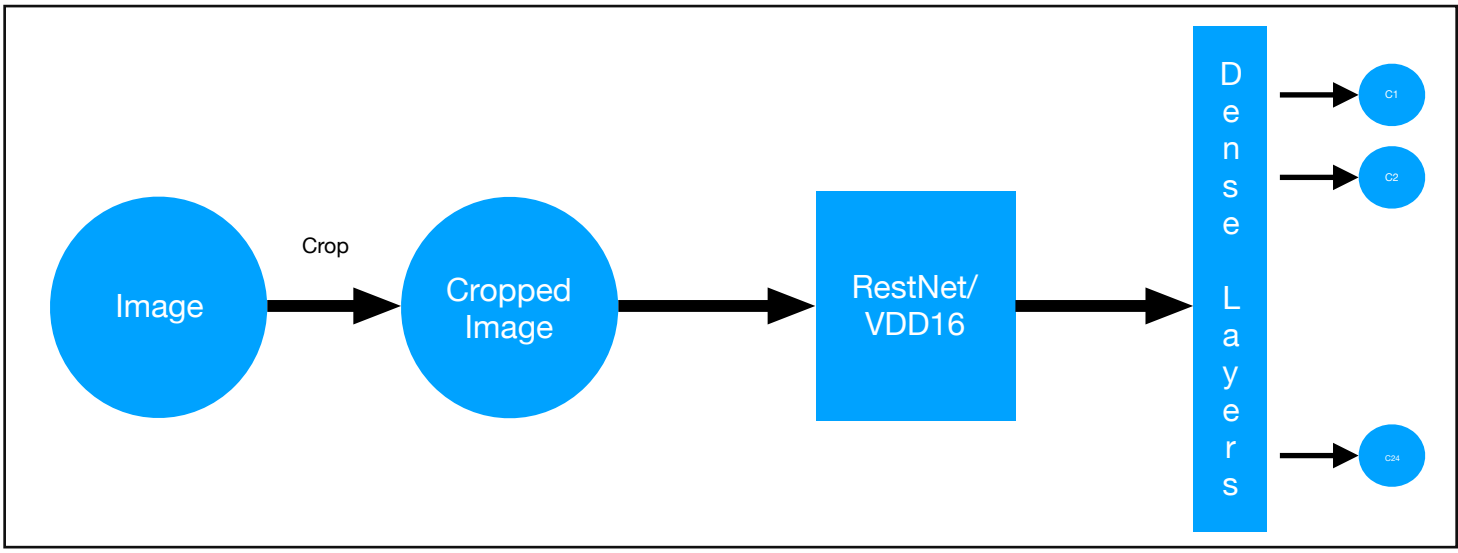

Figure 1: A sample network architecture for the proposed problem

The input to the network will be the training-dataset file which has images in it. The image is then cropped to include only the graphic by using techniques of openCV and its converted to a lower resolution equivalent. The image is then fed to the rest of the network and the output is then one hot encoded into one of graphic type attributes belonging to any of the 24 pre-defined categories.

[1] *derived column*

Benchmark Model

A basic Conv2D model with AveragePooling layer followed by 2 dense layers at the end, out of which the output layer will have 24 output neurons to predict one of the categories.

Evaluation Metrics

The model will be evaluated using **categorical cross-entropy** as a loss function on the **test** data and the accuracy metric for this model will be the **percentage** of total images

$$P = (X \div T) \times 100$$

that the model guesses right once it is trained on the **train** dataset. To give it a formula:
where $X = \sum$ number of attributes guessed correctly,
$T = \sum$ records in the dataset

Project Design

WORKFLOW:

- The input to the project workflow is the **train** dataset.
- Images and other features are extracted from the dataset
- Images are cropped to the graphic area and is fed along with other features in the dataset to the pre-trained ResNet/VGG-16 model.
- The output from these models is then fed to a fully connected 2/3-layered network of neurons with one of the layers being the output layer having 24 neurons where each one corresponds to a pre-defined category.
- **Adam** optimiser will be used along with categorical **cross-entropy** as the loss function and **softmax/relu** as the activation function. Different optimisers/loss functions can be tried in case the above ones doesn't suffice the accuracy metric as expected.

Stack to be used:
Langauge: Python
Libraries: Tensorflow, numPy, pandas, Keras, openCV
Platform: Jupyter