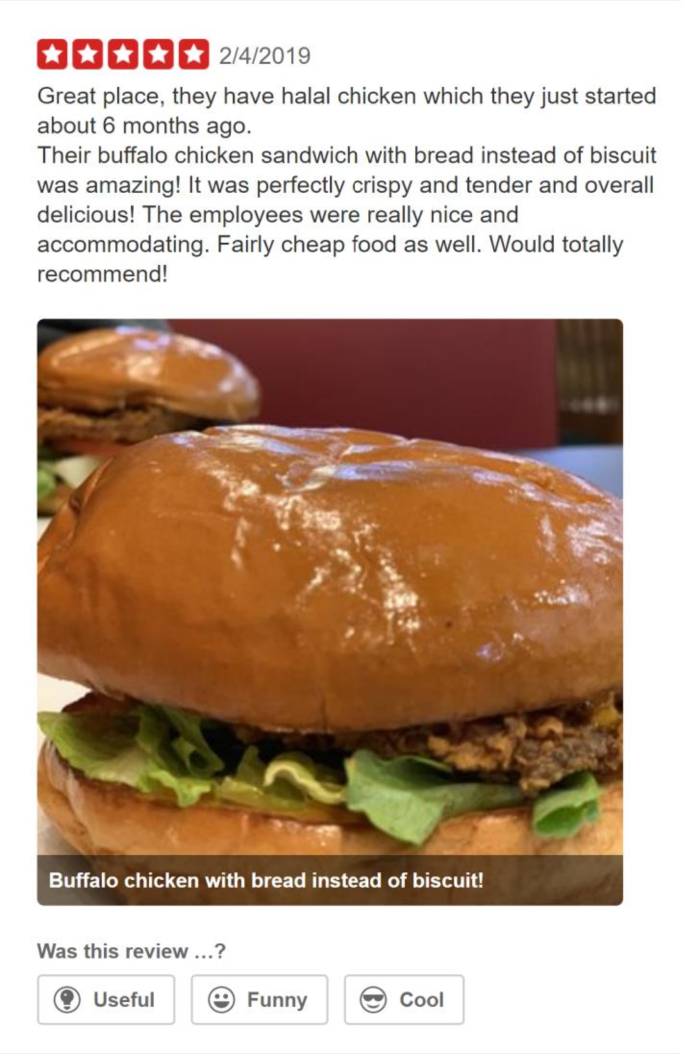
YELP RESTAURANT PHOTO CLASSIFICATION USING PYSPARK

Nishitha Chowdary (nxv170011)
Seshu Ram (sxs170832)
Shayan Sadar (sss170008)
Sneha Hulivan Girisha (sxh173730)

Index

Chapter	Contents	Page No.
1	Introduction and Problem Description	3
2	Related work	4
3	Dataset Description	5
4	Pre-processing techniques	6
5	Proposed Solutions and Methods	7
6	Experimental Results and Analysis	9
7	Conclusion	11
8	Contribution of Team Members	12
9	References	13

Chapter 1: Introduction and Problem Description



★★★★★ 2/4/2019

Great place, they have halal chicken which they just started about 6 months ago.
Their buffalo chicken sandwich with bread instead of biscuit was amazing! It was perfectly crispy and tender and overall delicious! The employees were really nice and accommodating. Fairly cheap food as well. Would totally recommend!

Buffalo chicken with bread instead of biscuit!

Was this review ...?

When you go to a new restaurant and have a good experience, it's likely that you won't keep the place a secret. After all, when people enjoy a great meal, exceptional service or a pleasant atmosphere, they typically make an effort to tell other people about it. Yelp is a social networking site that lets users post reviews and rate businesses. Founded in 2004 in San Francisco, Calif., the Web site is like a large online bulletin board featuring user-generated content, all geared toward personal reviews based on experiences at local businesses.

This dataset focuses the user submitted photos for reviews and have to be classified into labels. Currently, restaurant labels are manually selected by Yelp users when they submit a review. Selecting the labels is optional, leaving some restaurants un- or only partially-categorized.

Chapter 2: Related Work

Yelp Challenge has been hosted in the past as well. There have been 11 rounds of challenges so far. Last year Nicholas Egan, Jeffrey Zhang and Kevin Shen from Massachusetts Institute of Technology also worked with photo classification aspect of the reviews. They used Generative Adversarial Networks (GAN) to transform latent vectors drawn from a prior distribution into realistic looking photos. These latent vectors have been shown to encode information about the content of their corresponding images. They trained a GAN on food photos from the Yelp dataset and through several adjustments to the vanilla GAN formulation, managed to achieve photo-realistic results.

Also there is another work by Rajarshi Roy, who has also worked with photo classification with respect to the reviews of the customers, where more than thousands of images need to be classified and to build a model that automatically tags restaurants with multiple labels using a dataset of user submitted photos. Due to proven capabilities of the CaffeNet mode, a modified version of that was used. It has the same layers as that of AlexNet except for the fc8 layer and the output layer of the CaffeNet was modified for this project to suit the multi-outputs. Here, to measure the accuracy of the photo label classifier, four fold cross validation was performed which consisted of a set of training and testing images.Using the above methods, they could achieve an accuracy of up to eighty percentage and may be more parameters or architectures can be tried.

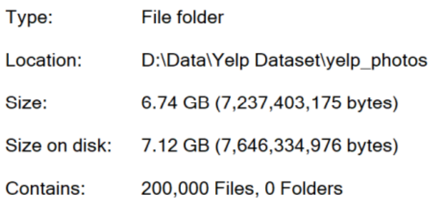
Chapter 3: Dataset Description

At Yelp, there are lots of photos and lots of users uploading photos. These photos provide rich local business information across categories.

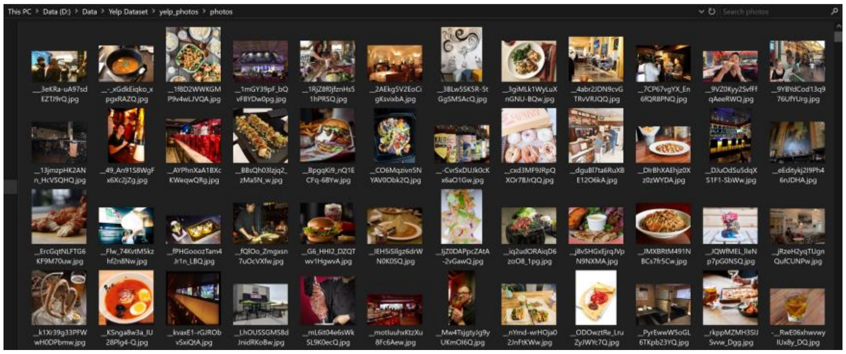
In this dataset, we are given photos that belong to a business and asked to predict the business attributes. There are 5 different attributes in this problem:

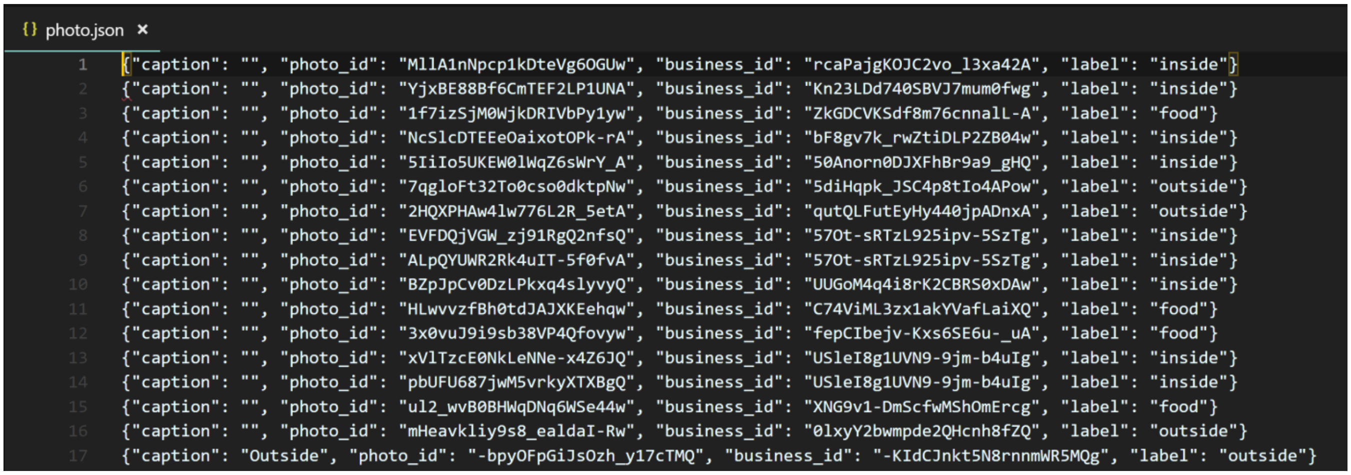
1. Food
2. Drink
3. Menu
4. Inside
5. Outside

The dataset consists of a folder of 200,000 images in random order that is about 7 GB. It also contains a .json file that consists of attributes for each image.



Type:	File folder
Location:	D:\Data\Yelp Dataset\yelp_photos
Size:	6.74 GB (7,237,403,175 bytes)
Size on disk:	7.12 GB (7,646,334,976 bytes)
Contains:	200,000 Files, 0 Folders

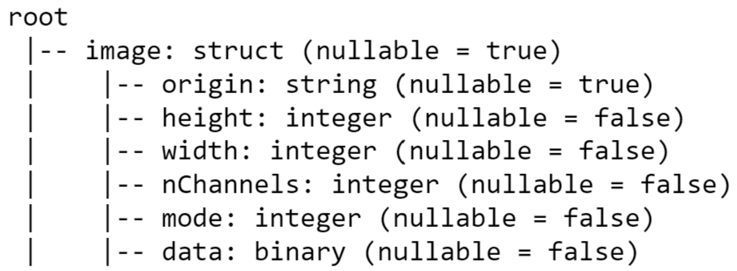

This PC > Data (D:) > Data > Yelp Dataset > yelp_photos > photos
Search photos



```
{"caption": "", "photo_id": "MllA1nNpcp1kDteVg6OGUw", "business_id": "rcaPajgKOJC2vo_l3xa42A", "label": "inside"}
{"caption": "", "photo_id": "YjxBE88Bf6CmTEF2LP1UNA", "business_id": "Kn23LDd740SBVJ7mum0fwg", "label": "inside"}
{"caption": "", "photo_id": "1f7izSjM0WjkDRIVbPy1yw", "business_id": "ZkGDCVKSdf8m76cnnalL-A", "label": "food"}
{"caption": "", "photo_id": "NcSlcDTEEeOaixotOPk-rA", "business_id": "bF8gv7k_rwZtiDLP2ZB04w", "label": "inside"}
{"caption": "", "photo_id": "5IiIo5UKEW0lWqZ6sWrY_A", "business_id": "50Anorn0DJXFhBr9a9_gHQ", "label": "inside"}
{"caption": "", "photo_id": "7qgloFt32To0cso0dktpNw", "business_id": "5diHqpk_JSC4p8tIo4APow", "label": "outside"}
{"caption": "", "photo_id": "2HQXPHAw4lw776L2R_5etA", "business_id": "qutQLFutEyHy440jpADnxA", "label": "outside"}
{"caption": "", "photo_id": "EVFDQjVGW_zj91RgQ2nfsQ", "business_id": "57Ot-sRTzL925ipv-5SzTg", "label": "inside"}
{"caption": "", "photo_id": "ALpQYUWR2Rk4uIT-5f0fvA", "business_id": "57Ot-sRTzL925ipv-5SzTg", "label": "inside"}
{"caption": "", "photo_id": "BZpJpCv0DzLPkxq4slyvyQ", "business_id": "UUGoM4q4i8rK2CBRS0xDAw", "label": "inside"}
{"caption": "", "photo_id": "HLwvvzfBh0tdJAJXKEehqw", "business_id": "C74ViML3zx1akYVafLaiXQ", "label": "food"}
{"caption": "", "photo_id": "3x0vuJ9i9sb38VP4Qfovyw", "business_id": "fepCIbejv-Kxs6SE6u-_uA", "label": "food"}
{"caption": "", "photo_id": "xVlTzcE0NkLeNNe-x4Z6JQ", "business_id": "USleI8g1UVN9-9jm-b4uIg", "label": "inside"}
{"caption": "", "photo_id": "pbUFU687jwM5vrkyXTXBgQ", "business_id": "USleI8g1UVN9-9jm-b4uIg", "label": "inside"}
{"caption": "", "photo_id": "ul2_wvB0BHWqDNq6WSe44w", "business_id": "XNG9v1-DmScfwMShOmErcg", "label": "food"}
{"caption": "", "photo_id": "mHeavkliy9s8_ealdaI-Rw", "business_id": "0lxyY2bwmpde2QHcnh8fZQ", "label": "outside"}
{"caption": "Outside", "photo_id": "-bpyOFpGiJsOzh_y17cTMQ", "business_id": "-KIdCJnkt5N8rnnmWR5MQg", "label": "outside"}
```

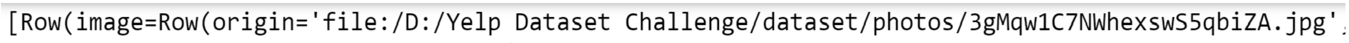
Chapter 4: Pre-Processing Techniques

We first loaded the images and converted into a dataframe. The schema of the dataframe is as follows:



```
root
 |-- image: struct (nullable = true)
 |    |-- origin: string (nullable = true)
 |    |-- height: integer (nullable = false)
 |    |-- width: integer (nullable = false)
 |    |-- nChannels: integer (nullable = false)
 |    |-- mode: integer (nullable = false)
 |    |-- data: binary (nullable = false)
```

We had to extract the file name from the origin column and match it with each label component in the .json file to performed supervised classification.



```
[Row(image=Row(origin='file:/D:/Yelp Dataset Challenge/dataset/photos/3gMqw1C7NWhexswS5qbiZA.jpg',
```

We did this by splitting the file name on '/' initially and then on ',' to get the filename without the extension. The .json file was then loaded into the dataframe and both the data frames were joined on the label name. This way we obtained the output column for the set of images in one single dataframe which was then ready to be processed and trained.

The dataset was split into 60:40 as the train and test data respectively.

Chapter 5: Proposed Solutions and Methods

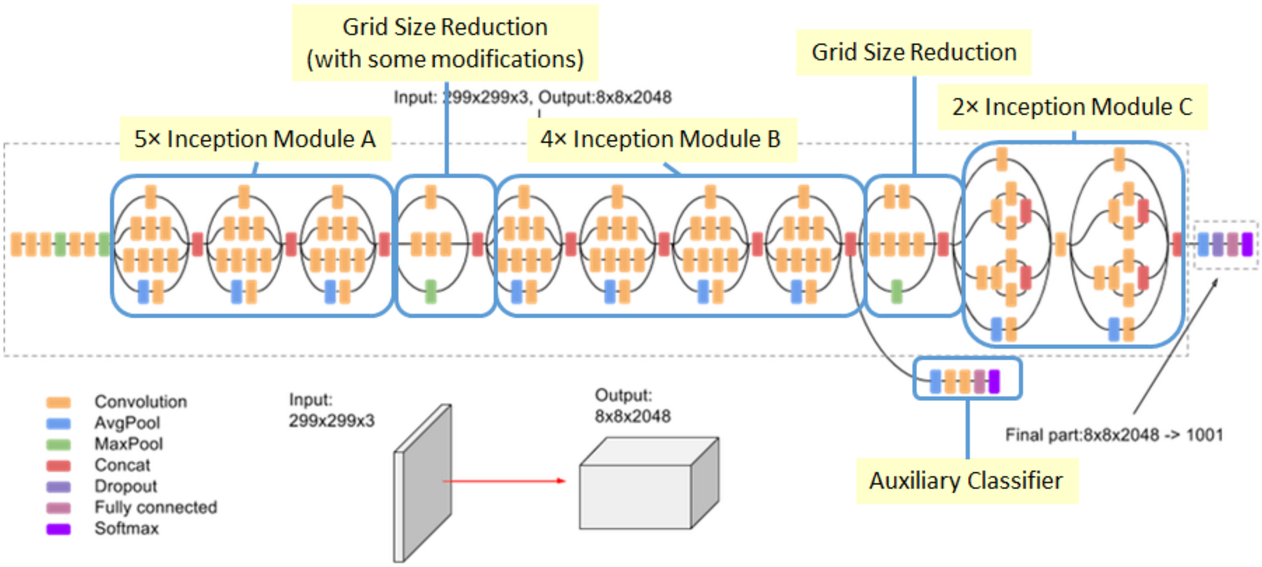
PySpark was used for the implementation of this project. Pyspark.ml and pyspark.sql libraries were used in the code.

Proposed Techniques:

It was decided based on our research that Transfer Learning technique was the best choice for the purpose of classification in this project.

When we consider classifying images, we often opt to build our model from scratch for the best fit, we say. This is an option but building a custom deep learning model demands extensive computation resources and lots of training data. Moreover, there already exists models that perform pretty well in classifying images from various categories.

One such model is Inception V3 by Google is the 3rd version in a series of Deep Learning Convolutional Architectures. Inception V3 was trained using a dataset of 1,000 classes from the original ImageNet dataset which was trained with over 1 million training images, the Tensorflow version has 1,001 classes which is due to an additional "background' class not used in the original ImageNet. Inception V3 was trained for the ImageNet Large Visual Recognition Challenge where it was a first runner up.



Transfer learning allows you to retrain the final layer of an existing model, resulting in a significant decrease in not only training time, but also the size of the dataset required.

We have 5 output labels for classification as the final label and Logistic Regression is used to perform this task. Logistic regression is a probabilistic classification model which is what we need to determine the output label On their own, logistic regressions are only binary classifiers, meaning they cannot handle target vectors with more than two classes. However, there are clever extensions to logistic regression to do just that. In one-vs-rest logistic regression (OVR) a separate model is trained for each class predicted whether an observation is that class or not (thus making it a binary classification problem). It assumes that each classification problem (e.g. class 0 or not) is independent.

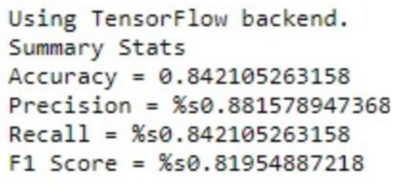
$$S(x) = \frac{1}{1 + e^{-x}}$$

Methods:

1. The dataset is stored in a S3 bucket on AWS.
2. Using Pyspark SQL, the name of the file extracted and stored in a dataframe.
3. This filename was then matched with the .json *'photo_id'* which has the true labels for each image and is stored in a dataframe.
4. Each label is assigned a numerical value as the labels.
5. 60% of the data is used for training and 20% was used for testing.
6. The labeled data is passed to the InceptionV3 model for training which is imported from the sparkdl library.
7. Logistic Regression is used as the final layer for the purpose of multiclass classification.
8. The accuracy metric is used to evaluate the predicted and true labels.

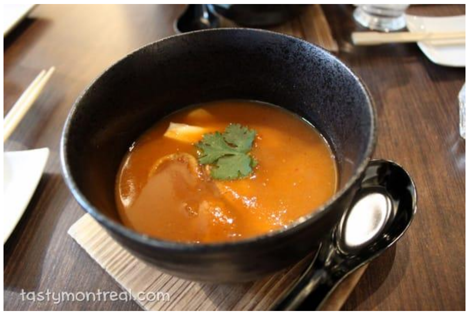
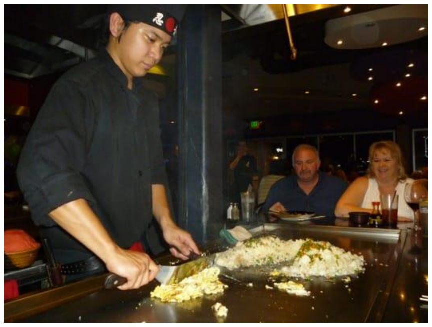
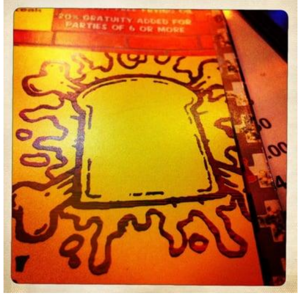
Chapter 6: Experimental Results and Analysis

Results:



```
Using TensorFlow backend.
Summary Stats
Accuracy = 0.842105263158
Precision = %s0.881578947368
Recall = %s0.842105263158
F1 Score = %s0.81954887218
```

Outputs:

Image	Actual	Predicted
	Food	Food
	Food	Inside
	Inside	Inside

	Drinks	Drinks
	Outside	Outside
	Menu	Menu

Chapter 7: Conclusion

We can conclude that Convolutional Neural Network (CNN) is one of the best methods for our problem of photo classification, which can be said looking at the results with an accuracy of about 84%. We have restricted our final layer to logistic regression, which could have been done with Gradient Boosting for more accurate results, but will take a lot of time due to the vast number of images present as it is computationally expensive. Another way of increasing the accuracy is by running it on more number of training images, but the resources we had were limited. Hyper-parameter tuning can be done on more powerful machines that have the capability to handle the computation and time cost.

Chapter 8: Contribution of Team Members

Each team member participated in the conceptualization and research aspect of the final project. We each read and shared a variety of different related papers regarding the topics of Photo Classification and went through the documentation of PySpark.

With regards to implementation, Sneha and Nishitha helped with the pre-processing of the data and obtaining the labels from the .json file and matching with the each photo. Seshu and Shayan worked on designing the Pipeline and using Machine Learning techniques to design the model for training the dataset.

Sneha and Seshu then setup the AWS cluster to train the model for various subsets of the dataset and obtain results.

Chapter 9: References

i.) https://spark.apache.org/docs/latest/api/python/index.html

ii.) https://spark.apache.org/docs/latest/

iii.) http://cs231n.stanford.edu/reports/2016/pdfs/014_Report.pdf

iv.) https://hackernoon.com/bag-of-tricks-for-image-classification-with-convolutional-neural-networks-paper-discussion-693c9e17d1cc

v.) https://pythonhosted.org/scriptine/intro.html

vi.) https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c

vii.) https://medium.com/datadriveninvestor/building-an-image-classifier-using-tensorflow-3ac9ccc92e7c

viii.) Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS’12), F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 1. Curran Associates Inc., USA, 1097-1105