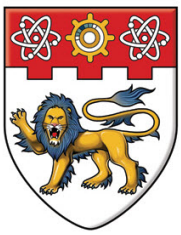

NANYANG
TECHNOLOGICAL
UNIVERSITY

CI6227: Data Mining:

Project Report on Kaggle Challenge

Contents

List of Figures

Member's contribution

Michele Barbera *was mainly involved in the Presentation for the lecture on the 2nd of November and the Report writing. Additionally this member implemented supporting features for the Food/Non-food classifier.*
Activity Level of the member: ***Active***

Jan-Eric Egenolf *was mainly active on the implementation of the Food-Non Food Classification model and support activities on TensorFlow Convolutional Neural Network implementation.*
Activity Level of the member: ***Active***

Christian Dallago *was mainly active both on the implementation of the Food-Non Food Classification model and TensorFlow Convolutional Neural Network implementation.*
Activity Level of the member: ***Active***

Yang Fan *was hardly providing support activities for the project. The member gave his contribution to the Presentation for the lecture on the 2nd of November.*
Activity Level of the member: ***Docile***

Evaluation Score

The level of accuracy gained is **64.3%** as shown in the Figure 1. The position reached is **181** which resides in the **Top 60%**.

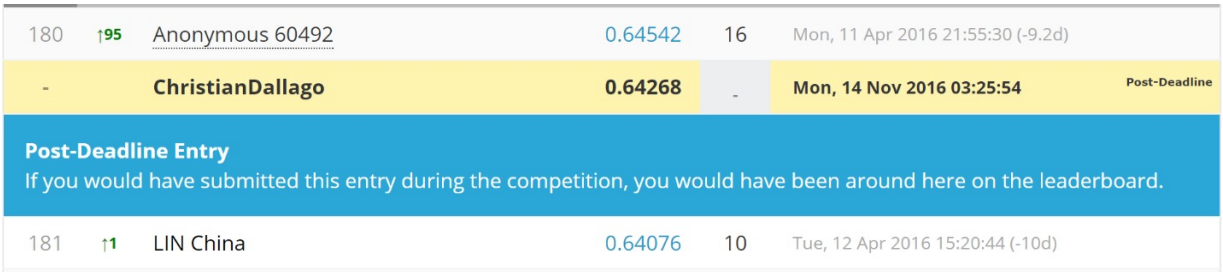

Figure 1: Results

1 Introduction

Nowadays, one of the most powerful source of information, whenever people choose which restaurant to go to and have a meal, are pictures. In fact, restaurants' owners that show photos of their places appear to customers as more trustworthy and reliable. Consequently, most of the restaurants today have to post photos on online platforms, in order to inform their customers about the services they can provide and the facilities clients can benefit from.

As a result, during the last decade many platform were developed in order to respond to the emerging trend, among the others Yelp was founded in 2004. Yelp is a free online platform, where business owners (or managers) can setup a free account to post photos and messages their customers (Yelp 2016, Yelp, 2016). Yelp provides a common ground for restaurant's manager to advertise their businesses at a relatively cheep price and for customers to pick thoroughly the place for their needs.

During the last 10 years, the success of Yelp service led the company to manage huge amount of data which had to be processed manually. As a matter of fact, pictures uploaded by customers and restaurants' managers had to be processed by Yelp human employees, who had to assign different labels (or tags) to the pictures received in order to have for every restaurant a list of tags that represent characteristics that the restaurant has. In order to remain efficient dealing the increasing amount of pictures uploaded by the users, in December 2015, Yelp published a challenge in Kaggle using the power of crowd sourcing given by Kaggle users.

2 Problem Definition

In the challenge published by Yelp, it is asked to solve a multi-label classification problem predicting the labels attributed to a given input picture among 9 most common labels:

- 0: good_for_lunch
- 1: good_for_dinner
- 2: takes_reservation
- 3: outdoor_seating
- 4: restaurant_is_expensive
- 5: has_alcohol
- 6: has_table_service
- 7: ambience_is_classy
- 8: good_for_kids

It goes without saying that this specific problem is a multi-label classification problem, since to one picture can be assigned different labels among the ones stated above Pan 2016.

3 Project Execution Framework

Our Project execution design concerned the use of GitHub, an online repository service that allowed the group to sync the project sections, update the code efficiently and to discuss internal issues. As a support, Python was used as a common coding language together with Jupyter Notebook, a server-client application that allows running notebooks documents via web browser. In addition to that, weekly meetings were arranged in order to discuss latest implemented features.

3.1 TensorFlow

TensorFlow is a is an Open Source Software Library for Machine Intelligence that allowed the team to use Convolutional Neural Networks (CNNs). CNN is one kind of feedforward neural network which is very efficient in finding patters and using them for classification. CNNs are applied in several areas: sound analysis, text recognition and most importantly, image recognition. CNN algorithms are a multilayer perceptron that is specialized in the identification of two-dimensional image information. These networks have usually more than one layer: input layer, convolution layer, sample layer and output layer. All these layers are stack into a process that transform the pictures at every layer (or step), identifying the main characteristics of the image, that determine in the end, the distinguishing aspects that drive the prediction of the machine into one output or another.

Architecture of TensorFlow

Representing Tensors In the real world, a convenient way to describe an object is to list out its properties. Or put it in machine learning way, its features. This ordered list of features is referred to as the feature vector. Each data item normally consists one feature vector, as depicted in the figure below. With a dataset of thousands of data, feature vector can grow up to thousands. These feature vectors are collectively represented as a matrix 2.

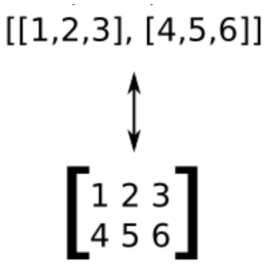

Figure 2: Vectors represented as matrix (1)

In the TensorFlow context, matrices are represented as vector of vectors. A tensor is a way of generalizing the matrix that specifies an element by an arbitrary number of indices 3. The syntax for tensor is a nested vector. A tensor's rank is the number of indices required to specify an element.

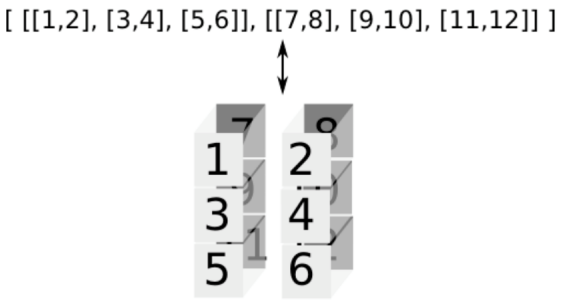

Figure 3: Vectors represented as matrix (1)

Executing operation with Sessions A session is a software system environment that describe how the software system should run. In TensorFlow, a session is the interface between the hardware and the programming language interface (Python). Typically, user defines operations and algorithms in Python, send it to a session, and TensorFlow will call its underlining computation library (C++) to talk to hardware and do the heavy lifting N. 2016.

How tensor flows In TensorFlow, every operation can be considered as a node in a graph, as shown in the figure below. The edges between operators represent compositions of math

functions. For example, the neg operator is a node in the graph; the inbound and outbound edges of the neg node are how the tensor flows (Hence the name, tensor flow). The figure 4 shown here represents a Gaussian function "TensorFlow API Documentation" 2016.

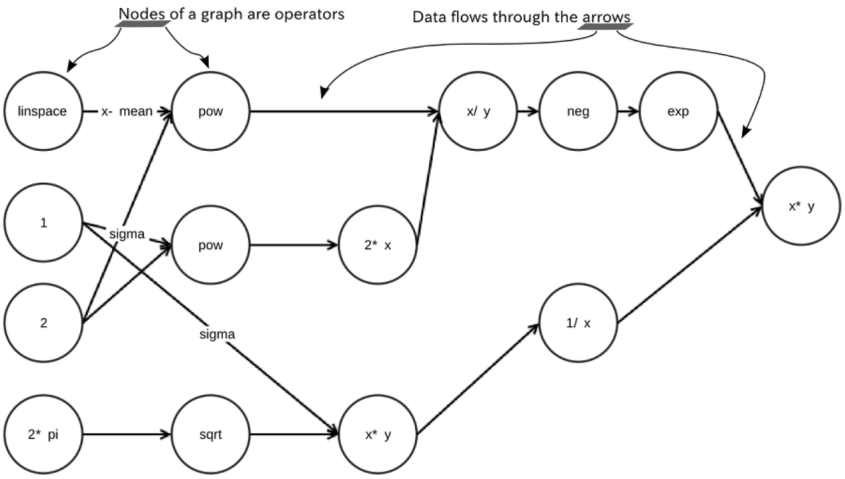

Figure 4: Gaussian Function

Although operations are represented in graphs, the actual computation must be run by a session 5. In addition to running graphical computations, session can also take input types like placeholders, variables and placeholders:

- Placeholders: An unassigned variable, but will be assigned in future operation
- Variable: A mutable variable object
- Constant: An immutable variable object

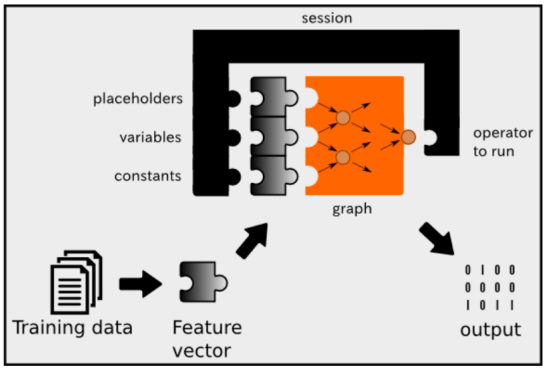

Figure 5: TensorFlow session

4 Challenges of the Problem

The Kaggle Yelp Photo Classification problem was since the beginning quite challenging and time-demanding. Throughout the whole project the team faced few but important challenges which are presented in the following sections.

Data handling The first main challenge faced by the team was the data handling activities. The main reason behind it was that the input data are represented by pictures, therefore the team had to deal with unstructured data that is not organized in a pre-defined manner RN1. Furthermore, the team had to cope with picture processing features and picture recognition tools, a new horizon for all the members in the group. On the top of that, the heavy load of the files' size made the handling of the data set more complicated. One of the main ground based feature that has been used during the project is the conversion from RGB (Red, Green, Blue) pictures, which are the normal colorful pictures, from Grey scale. In fact, while an RGB picture is a set of 3 matrixes which might be hard to handle, Grey scale pictures, on the other hand, can be represented by only one matrix.

5 Methodology

5.1 Our Approach

Our Approach concerned a # step process which is preceded by Data Cleaning Process described below. In the figure 6 is described the three different approaches implemented:

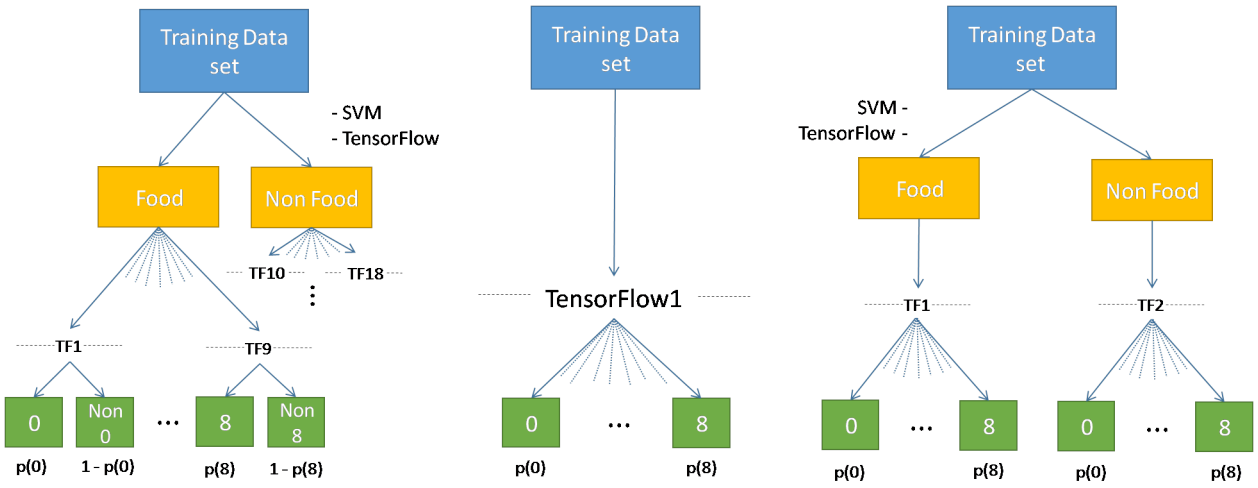

Figure 6: Approaches' Structure

5.1.1 First Approach

The First approach concerned the use of Food vs Non-Food Classification as a first step either with an SVM or with a Convolutional Neural Network implemented in TensorFlow. Following up, 18 different Convolutional Neural Networks were trained, in this approach, each model predicted the probability *p* that the picture, already classified as food or non food, was connected to a specific label and consequently the probability *1-p* that the label was not assigned to the picture.

5.1.2 Second Approach

The second approach concerned a direct training of a Convolutional Neural Network, implemented in TensorFlow, without prior Food vs Non Food distinction. The model would give a direct probability of which label has the higher probability to be assigned on a picture.

5.1.3 Third Approach

The Third approach resemble the second one, the only difference is that prior to the implementation of a Convolutional Neural Network on TensorFlow, a Food Vs Non Food Classification is implemented. The structure shows the first step as the Food vs Non-Food classification.

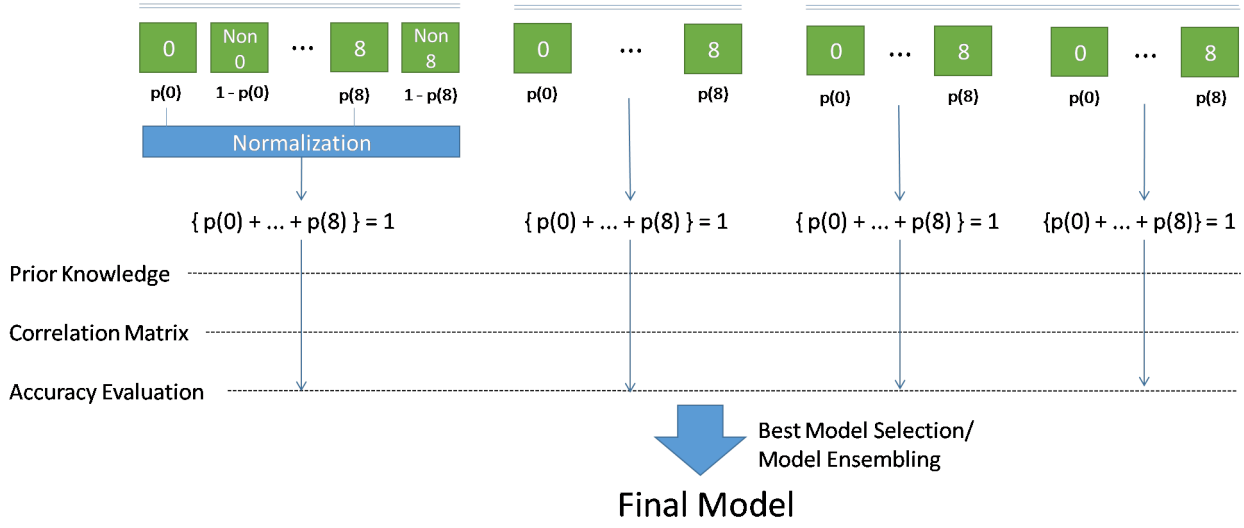

Figure 7: Following Approaches Structure

Following up to the three approaches, some features to the model were added. Particularly, in the three approaches, Prior Knowledge and Correlation Matrix information (further explained in Section 5.1.3) were applied. Finally the Accuracy evaluation would give the result of the best approach. In addition, Model Esembling practices were considered.

Correlation Matrix In order to improve the performance of the classifier, information from Correlation Matrix were considered. Particularly, Figure **??** shows the correlation between the labels. Three pictures were taken of the matrix, showing the probability *p(x,y)* that a label x would be assigned to a business, given that that business has already label y assigned (e.g. 'table-service' (label 6) is more correlated with 'serving alcohol' (label 5) than with the restaurant being 'good for lunch' (label 0)).

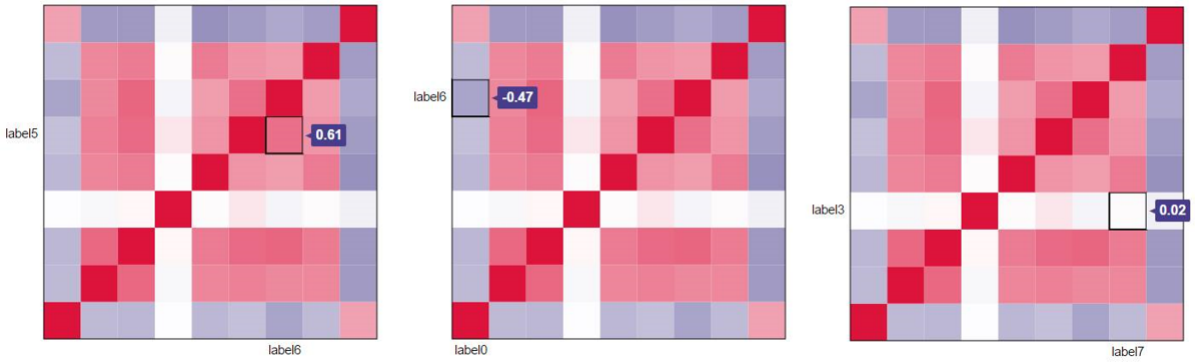

Figure 8: Correlation Matrix

Prior Knowledge Another kind of information that can be used, once the three kind of approaches have been developed is the Frequency of Labels in the Pictures and Businesses. Particularly, the frequency of the Labels both from the pictures and from the businesses is shown in Figure 9.

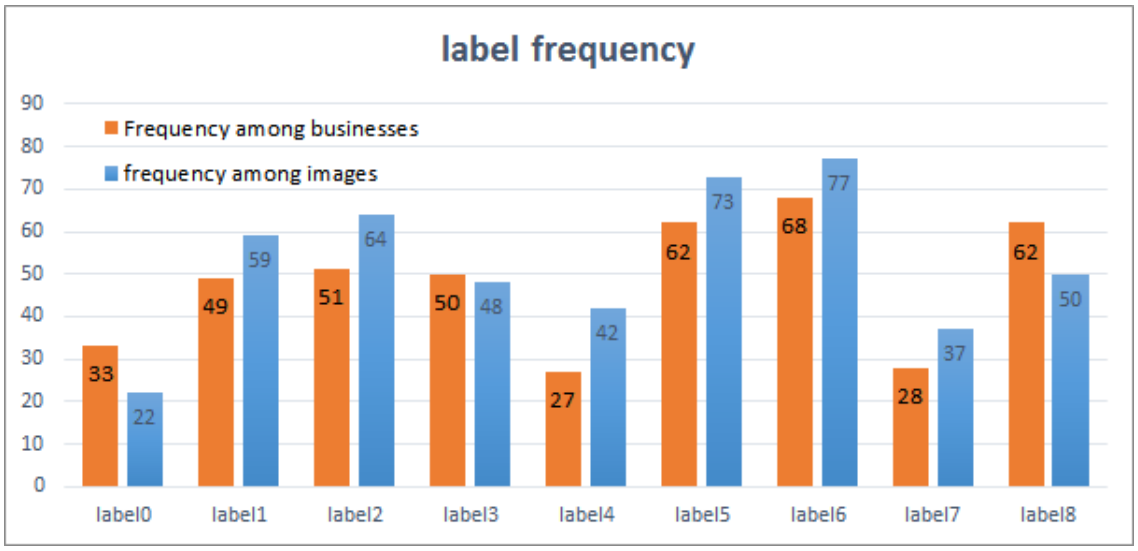

Figure 9: Label Frequency

5.2 Data Cleaning

The team's approach concerned several steps in a structured road map. The first steps of the project concerned mainly data cleaning activities (Duplicate and Normalization). Following up the data had been processed using different machine training activities. The following sections present the full road map of what has been done for the project in chronologically ordered.

5.2.1 Duplicate Identification

One of the first operation that has to be done on the data set, is the cleaning process. In regards to that the team has faced the first challenge, which was the identification of the duplicates in the data training set. Our approach concerned 4 main steps:

1. Transform images from RGB to Grey value
2. Resize of Images
3. Create of an hash for the image and map to file name
4. Compare hashes and assign label to file

The output of the cleaning process gave a set of pictures without duplicates. The original set, consisted of 234 842 pictures, after the cleaning process, the set reduced to 232 432 pictures, meaning that the duplicates accounted for roughly 1% of the data set. It is important to point out in this case that the identification and elimination of the duplicates was not crucial for the analysis. In fact, the noisy data accounted for just a negligible portion of the dataset. However, in order to perform a thorough analysis, this feature of duplicates removal has been considered.

5.2.2 Normalization

Once the duplication removal process has been completed, the next step concerned the Normalization of data. In fact, the pictures that we took as an input had both completely different dimensions and different shape. Consequently, the team produced a code that was cropping the pictures rescaling them. The Cropping feature was coded so that the pictures were cropped keeping the center of the image and cropping the longer edge so that the image become a square. At this point all the pictures were reshaped as squares of different dimensions. Once the first part was completed the pictures were resized into the same size. Particularly, two different batches of different size were kept in order to train the machine with different batches of data. The size of the picture kept for further processes are: 100 x 100 and 300 x 300 pixels.

5.3 Food/Non-Food Classification

The next step in the process was the classification between Food pictures and Non-food pictures. In fact, the data given included not only, pictures of food but also pictures showing other aspects of the restaurant (e.g. the ambient, the parking lot, the owners etc.). Therefore, it was crucial to classify the pictures according to this difference.

5.3.1 SVM Approach

The first approach include the usage of a Support Vector Machine (SVM) algorithm. Particularly, different configuration has been tried, varying the parameters and using different datasets. Once the best value for the parameters have been identified, the results have been collected.

The Figure 10 shows the structure for training the machine. 5 SVMs were trained belonging from a data set of 1000 samples. Each SVM included 1/5 of Data set for Cross Test.

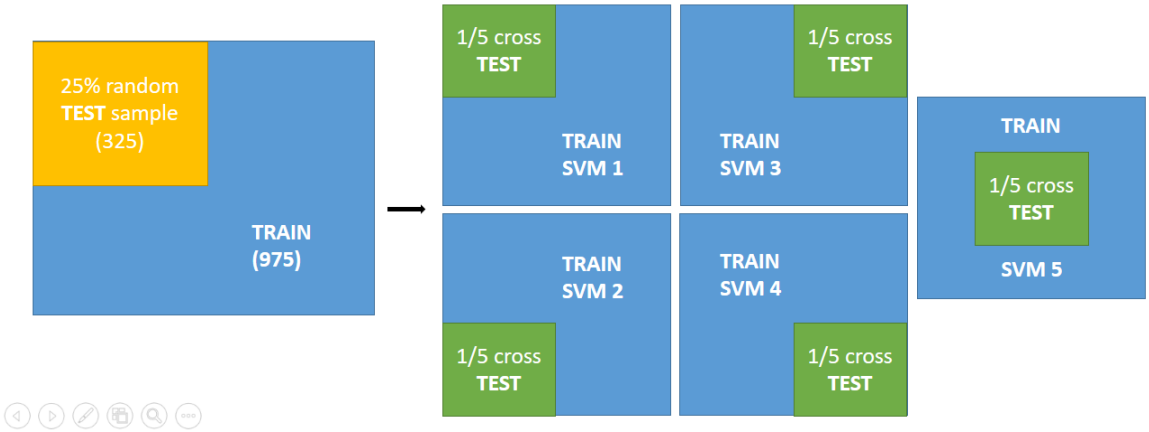

Figure 10: Food Vs Non-Food Classification Structure

The second step that has been tried was the creation of new features in order for the SVM to classify the images more accurately. Particularly, 5 different features presented hereby, has been coded Scikit-image is an image processing toolbox for SciPy:

- Skeletonize
- Watershed segmentation
- Edge Operators
- Contour Finding
- Binarization

Skeletonize Skeletonization reduces binary objects to 1 pixel wide representations. The algorithm works by making successive steps on the image. On each pass, border pixels are identified and removed on the condition that they do not break the connectivity of the corresponding object (Scikit-image 2016).

Watershed The watershed is a classical algorithm used for segmentation, that is, for separating different objects in an image. The watershed algorithm treats pixels values as a local topography (elevation). The algorithm floods basins from the markers, until basins attributed to different markers meet on watershed lines. In many cases, markers are chosen as local minima of the image, from which basins are flooded (Scikit-image 2016).

Edge Operators Edge Operators uses edge detection algorithms. Particularly, they are discrete differentiation operators, computing an approximation of the gradient of the image intensity function. Different operators compute different finite-difference approximations of the gradient. For example, the Scharr filter results in a less rotational variance than the Sobel filter that is in turn better than the Prewitt filter. Figure 11 shows the example of Edge Operators applied on a record of the data set (Scikit-image 2016).

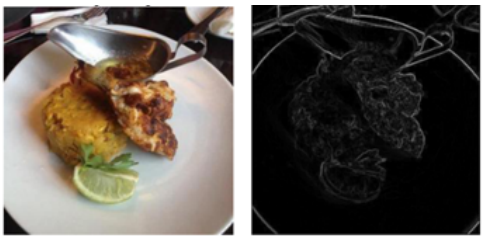

Figure 11: Edge Operators Example

Contour Finding The algorithm uses a marching squares method to find constant valued contours in an image. Array values are linearly interpolated to provide better precision of the output contours. Figure 12 shows the example of Contour Finding applied on a record of the data set (Scikit-image 2016).

Figure 12: Contour Finding Example

Binarization The algorithm binarizes an image using the $threshold_adaptive$ function, which calculates thresholds in regions of size $block_sizesurroundingeachpixel(i.e.localneighborhoods).Eachthresh$ $image2016$).

Figure 13: Binarization Example

The results from the Support Vector Machine were poor and inconsistent. Hereby, Figure 14 shows the best results obtained with this approach.

	precision	recall	f1-score	support
not-food	0.44	0.44	0.44	**62**
food	0.87	0.87	0.87	**263**
avg / total	0.78	0.78	0.78	**325**

Figure 14: SVM Results

5.3.2 Convolutional Neural Network Approach

The second approach tried for the classification of Food Vs Non Food, was the use of Convolutional Neural Networks (CNNs). Particularly two layers of convolution were coded for the network, together with two layers of Rectified Linear Units. The approach used concerned the construction of a cost function and its sequential decrease according to the iterations.

One of the first problem experienced was the fact that the cost function was not decreasing, but instead assumed different and random values. The problem depended on the initialization of the variables. Particularly, variables such as the bias and the weights were, at first, initialized as 0. The second approach, concerned the initialization of the variables according to the Normal distribution centered in 0. Even in this case the cost function was not converging to a value. The last approach involved the truncated Normal distribution initializing the variables. In this last approach the cost function was smoothly converging to a minimal value.

Another feature introduced in the model was the introduction of the real proportion between *Food* and *Non-food* to train the machine. The reason was to avoid the machine to be trained on a un-balanced proportion.

6 Post Processing

6.1 Post Processing in the first Approach

Once the prediction of the 1 approach have been calculated the result is going to be a percentage of likelihood that the label is assigned to the picture. Here the first threshold is applied. Particularly, the threshold is set to 60% meaning that has 60% probability to be assigned to that picture is taken and goes further in the next step. The next step involves a normalization of the likelihoods so that they add up to 1. Additionally, this post-processing technique will involve another two labels. These two additional labels are the ones with the highest likelihood in the above step, not including the ones already taken into account (e.g. if $label_0$, $label_1$ and $label_2$ overcome the first 60% threshold, we consider another two labels, for example $label_3$ and $label_4$ that have the highest likelihood in the rank that does not include $label_0$, $label_1$ and $label_2$).

If after the normalization is shown that $label_1$, $label_2$ and $label_3$ are the best. Even if $label_3$ was not considered in the first step, it is taken into account anyway.

6.2 Correlation information in the post processing

The correlation of the labels gives a lot of information and can be crucial in the post processing for boosting up the performance of the classifier. Particularly, it has been calculated the likelihood of every combination of arrays with all the labels. Meaning *"how likely that an image has only* $label_0$ *"* or *"how likely that an image has both* $label_0$ *and* $label_1$ *"* and so on. The result is a number that tells which ones are the most likely and which were never observed. Some combination of labels, for example, were never experienced in the data set. As a result, it is unlikely that the same combination can be predicted and if it is predicted just one label has to be taken into account (the most likely one).

Adding this to the post processing activity above, it can be calculated the frequency of the combination between the first choice labels (e.g. in the example $label_0$, $label_1$ and $label_2$) that overcame the first threshold and the additional two labels (e.g. in the example $label_3$ and $label_4$). The highest number in the array of "frequencies" is going to give the final combination of labels assigned to the picture.

7 Results

This section presents the results obtained from the classifier, which are illustrated in a *.csv* file attached to this document. The final predictor, is the result of the overall process which has been presented above in this paper. Moreover, the prior knowledge has been included in the model.

Even if the results are not promising as expected, the team has put its effort into trying and implementing different approaches and features. The last level of accuracy reached from the model is 0.64298.

This level of accuracy resulted in an average position in the rank (181), which ensure the Top 60%.

8 Conclusion

This paper has presented the Kaggle Challenge: Yelp Photo Classification. This challenge has revealed many aspects that were quite difficult to handle, especially the data processing since images handling activities were quite new for everyone in the team.

Nevertheless, the team has worked on different approaches and features that presented different threats and problems. At the end of the project, even if the result were quite poor, the team is satisfied since it has overcome many challenges with discrete results.

References

N., Shukla. 2016. "Machine Learning with TensorFlow". *(MEAP ed.). Manning.*

Pan, Sinno Jialin. 2016. "Lecture 2a: Data". *CI6227: Data Mining.* `http://www.ntu.edu.sg/home/sinnopan`.

Scikit-image. 2016. "Scikit-image processing in python". `http://scikit-image.org/`.

"TensorFlow API Documentation". 2016. `https://www.tensorflow.org/versions/r0.11/api_docs/index.html`.

Yelp. 2016. "About Us". `https://www.yelp.com.sg/about`.