
Residual Neural Network and Text Analysis: Combined Methods in Image Ranking

Arindrima Datta, Kiyan Rajabi, Roy Cohen

Abstract

We present a novel approach to ranking images using a combination of analysis techniques that broadly encompasses natural language processing of descriptions, analysis of category tags, and investigation of features extracted from a residual neural network (ResNet) for a corpus of training images. In this framework, for any instance of image description of an unknown image, 20 images are returned in order of their relevance. The classification process commences with a Bag of Words processing of the query description, which is then transformed into a more compact low-dimensional space using dimensionality reduction techniques. The compact Bag of Words is mapped to the outcome space of ResNet features and text category tags using two separate computation pipelines. Finally, we use a K-Nearest Neighbor algorithm to determine the top 20 relevant images for the query.

1 Introduction

In 2013, Google acquired DNNResearch, a start-up created by Professor Geoffrey Hinton and his graduate students at the University of Toronto. They built "a system that used deep learning and convolutional neural networks and easily beat out more traditional approaches in the ImageNet computer vision competition designed to test image understanding [1]." Google built and trained similar large-scale models and found that this approach doubled the average precision, compared to other object recognition methods. "We took cutting edge research straight out of an academic research lab and launched it, in just a little over six months," says Chuck Rosenberg, from the Google Image Search Team.

Since then, neural nets have become the state of the art technology that Google uses to create tags for images in its system. Every day, hundreds of millions of people who access Google Images use words in order to access images through a tagging mechanism that lacks any human intervention. We live in a time in which the ability to search images is relevant to the lives of millions of people around the world.

And yet, the problem of image recognition is immensely complicated. Tagging mechanisms rely on high-dimensional data, in which hundreds of tagging criteria are used to describe millions of images. Single images are described only by several words. To successfully apply sparse representation to computer vision tasks, we have to address the problem of correctly representing the data [2].

As the field of neural networks develops, so do the possibilities image classification increase. Competitions in the field of computer vision such as the ImageNet Large Scale Visual Recognition Challenge bring together some of the best thinkers in Computer Science to tackle challenges in the field of computer vision. One of the results of these challenges has been *ResNet*, a state-of-the-art residual learning framework [3].

In this work, we develop a general framework for using image tags as well as feature vectors taken from *ResNet*'s residual network analysis to rank images in response to one-sentence descriptions.

We propose a model that processes the image data in two separate channels: one dedicated to word tags, and the other to feature vectors that pertain to each image’s visual elements. After performing dimensionality reduction and classification in each of the two analysis routes, we look for overlapping classification results between the two methods by performing a nearest neighbor search in their combined space. This allows us to map the unknown description to images in the image pool in order of their similarity of features and tags to the former.

2 Related Work

Due to the relative ease of understanding and processing text, commercial image-search systems have often relied on techniques that are largely indistinguishable from text search [4]. Image retrieval research focuses on the utilization of indexed image collections [5]. Researchers and practitioners have created various thesauri for visual information. Image system and indexers then use these thesauri to index images within a collection [6].

Recently, academic studies have demonstrated the effectiveness of employing image-based features to provide either alternative or additional signals to use in this process [6]. Using visual information to re-rank and improve text based image search results is an idea that has gotten traction in the computer vision community. In Traditional Neural Networks, each layer produces the desired output by calculating $y = f(x)$. With residual learning, proceeding layers of the deep neural network help alter the output from the previous layer. ResNet layers calculate $y = f(x) + \text{id}(x) = f(x) + x$ instead to introduce identity connections between them (See Figure 1) [7]. As such, the authors claim ResNet Neural Networks are easier to optimize and result in higher accuracy [2].

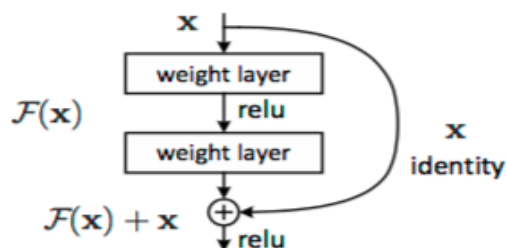


Figure 1: A Single Iteration in a Residual Neural Networks

In particular, it has been demonstrated that given a set of images returned by text based image search engines, visual features can be adaptively leveraged to re-rank the results [8].

3 Contribution of This Work

Our primary contribution in this work is a robust but simple process for combining text tags and RNN-induced features in optimizing a ranking procedure for images. The images used in this study are aligned with ones that users love to upload and search for, e.g. photos of food, pets, and people playing sports or engaging in comical activities [9]. In addition to building a sophisticated way of preprocessing the natural language descriptions, we also created a transparent mechanism by which to assess the distinct methods of text analysis and visual residual networks against each other.

4 Dataset

4.1 Motivation

Given a one sentence description of a scene, we would like to return the 20 most similar images out of a pool of 2,000 options.

4.2 Description of Data

The corpus of training data contained images, tags, descriptions, and ResNet feature vectors (see Table 1).

Images	10,000 images adhering to identical dimensions (224x224 pixels)
Text Tags	A set of text tags that correspond to each image (10,000 sets in total)
Descriptions	A set of five, one-sentence descriptions for each image (10,000 total description sets)
ResNet Feature Vectors	Two sets of feature vectors extracted for each image from <i>ResNet</i> , a pre-trained residual network. The two sets were extracted from: <ul style="list-style-type: none"> • Penultimate <i>pool5</i> layer (2048 feature per vector) • Final <i>fc1000</i> layer (1000 feature per vector)

Table 1: Training Data


Training Image	Training Descriptions	Training Tags
	<p>A red and blue fire hydrant and people walking in the rain.</p> <p>A fire hydrant on a wet street in a small city.</p> <p>A blue and red fire hydrant stands in a brick city square in the rain.</p> <p>A blue and red fire hydrant sitting on the side of a road.</p> <p>A blue fire hydrant on a brick road.</p>	<p>accessory:umbrella</p> <p>vehicle:car</p> <p>outdoor:fire hydrant</p> <p>person:person</p> <p>accessory:handbag</p> <p>vehicle:truck</p>

Figure 2: Our sample data included 10,000 square photos, 5 description sentences, and 1-7 tags that match the photos.

5 Approach and Algorithm

5.1 Preprocessing

5.1.1 Bag of Words

The Bag of Words model is one of the most popular representation methods for object categorization [10]. The key idea is that each description sentence is represented as a vector, in which each cell is the frequency of an individual word appearing in the text. For this purpose, we have merged all the descriptions for an image into a single text with the intuition that a word appearing in multiple descriptions should have more weight over less frequent words.

In order to reduce noise, we initially removed stop words (using Python NLTK's corpus of stop words) and stemmed all the words (using NLTK's SnowBallStemmer). Using cross-validation, we found that it was effective to remove certain parts-of-speech and ended up including the following in our model: nouns (common and proper nouns), adjectives, and certain types of verbs (base form, action verbs and gerunds).

As every sentence contains only a few words from the vocabulary, the bag of words representation was sparse. This ultimately affected our model because of the curse of dimensionality. Hence, we used dimensionality reduction to transform the data into a compact lower dimensional space using Singular Value Decomposition (SVD). SVD is a widely used technique to decompose a data representation into several component matrices, exposing many of the useful and interesting properties of the original matrix [11]. After performing cross validation on SVD's dimensions, we settled for an optimal dimension of 1,000.

5.1.2 ResNet

The features in the original dataset were extracted using ResNet, a technique that allows researchers to effectively train deeper neural networks to extract content out of images [2]. We obtained the ResNet data, which contained information about the images' residual neuron network analysis. For each image, we looked into two sets of multidimensional representations: 1,000 and 2,048 respectively. The features from penultimate layer of ResNet (2,048) comprised of a richer representation of the image, which we determined using cross-validation and visualization (Figure 3) and used this dataset for our analysis.

For Bag of Words, the high dimensionality of the ResNet data was affecting the performance of the model. To tackle this problem, we used singular value decomposition (SVD) to reduce dimensionality on data collected from residual networks [18]. We reduced the data to 60 dimensions. The dimensionality reduction of Bag of Words categories data and ResNet data are shown in Figure 4.

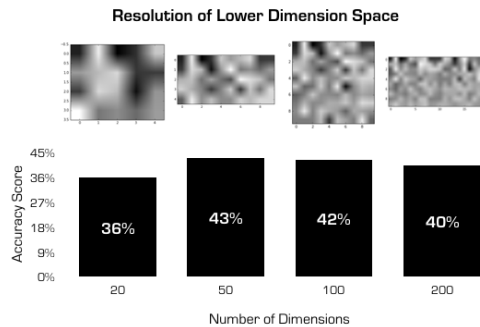


Figure 3: The ideal reduced dimensionality of ResNet data was found between 50-100. We used SVD to project data to lower dimensions. The grayscale images are visualizations of projected feature vectors.

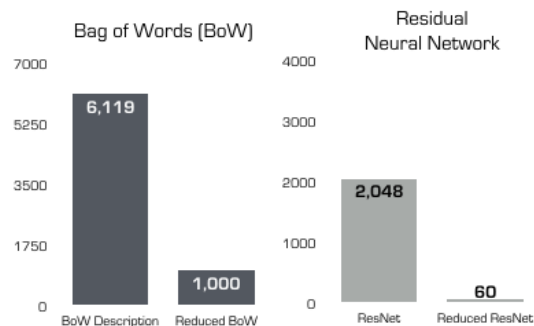


Figure 4: Dimensionality reduction took place across text and visual representations of images

5.1.3 Bag of Words for Tags

Each image included in our training pool had a set of tags associated with it. We used a binary bag of words representation of the categories associated with the images. Each vector had length 80, corresponding to the total number of categories or tags provided with the data.

5.2 Training of Classifier Models

A great deal of research on image ranking has been done using Bag of Words and regression models [12]. Following in that path, we tried different regression models and parameter optimizations.

We soon realized that we have two outcome spaces, inherently different in nature. The ResNet features were derived from a population of continuous variables, while the categories vector comprised of binary variables. Initially, we had combined these two outcome spaces into merged vectors and used a single regression method. In that case, we treated the dependent variable as continuous. Although this method worked to some extent, its scores were low, up to 0.21.

Therefore, we created two separate pipelines trained with different models and combined the results. In both cases, we used multi-output modelling strategies since our dependent variables (ResNet and Bag of Categories) are multi-dimensional.

In assessing the validity of our models, we performed repeated cross-validation and looked at the average score from the analysis. In cross validation, we utilized a 70-30 train-test split.

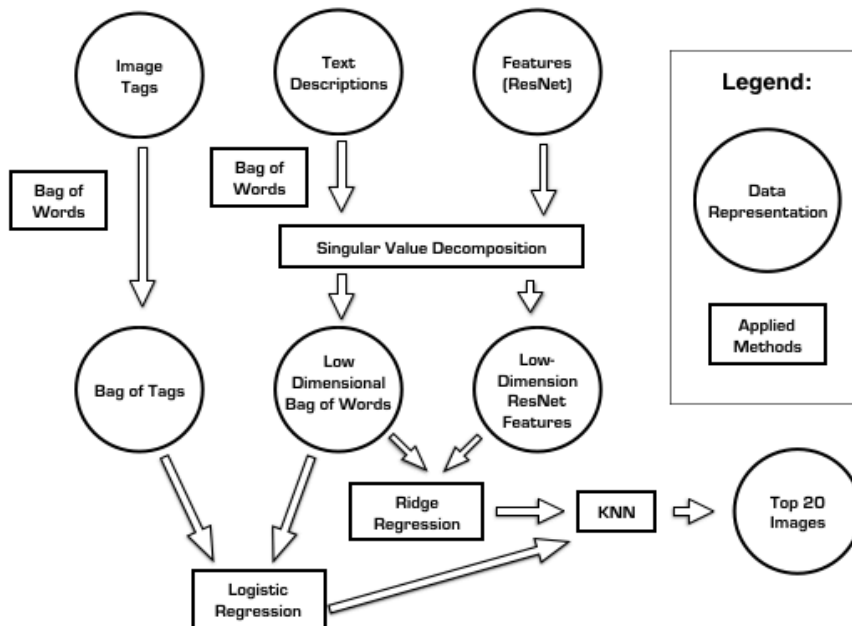


Figure 5: Illustration of Our Pipeline

5.2.1 Regression of Bag of Words with ResNet Features

For the ResNet Features, we tried several regression models, starting with the rather-simple KNN and eventually escalating to more complex models like Ridge and Lasso (Table 2, Appendix). We found that Ridge Regression with an alpha of 10 gave us the best regression score in cross validation (0.50), an effect that carried over to higher accuracy in prediction of ranking images.

Ridge Regression is an approach that has performed well in relevant studies in the past [20]. For Linear Regression, an output is predicted from inputs and assigned weights. Ridge Regression penalizes overgrown weights, thus preventing individual features from being weighted too significantly [13]. This technique is called regularization. We used Ridge Regression in order to map the descriptions' Bag of Words model onto the ResNet features.

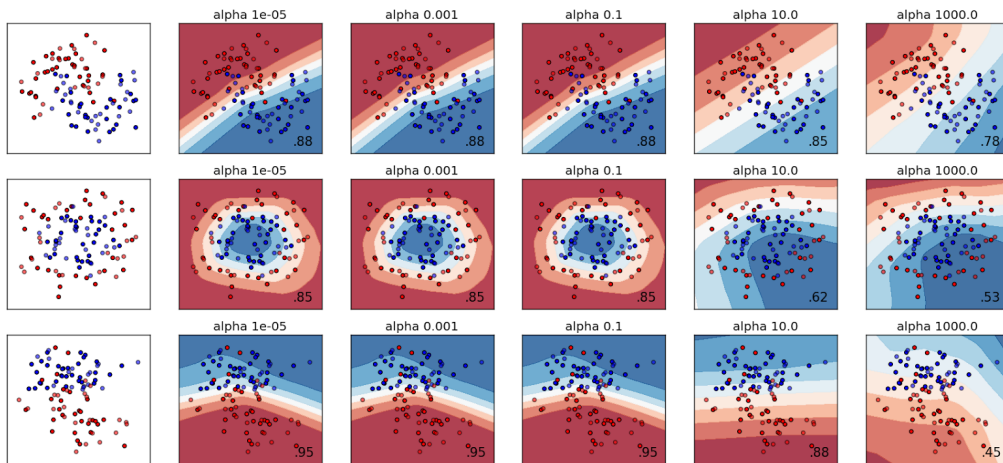


Figure 5: A comparison of different values for regularization parameter ‘alpha’ on synthetic datasets. Alpha is a parameter for regularization term (penalty term) that combats overfitting by constraining the size of the weights. [25]

5.2.2 Regression of Bag of Words with Bag of Categories

Simultaneously with the regression of the descriptions onto the visual vector features, we also conducted a regression onto the one-word category space. Text-based analysis is salient to image search, and the category space provided a rich platform of data which we could utilize [16]. The Bag of Categories contained only binary values; after testing different classifiers, we eventually used logistic regression to classify the descriptions onto the category space.

Using cross-validation, we found out that Logistic regression returned an accuracy rate of 0.51 among all the models we have tried (Table 2, Appendix).

5.3 Searching the Image Space for the Final Result

Our goal was to return 20 images (in order of relevance) in response to a one-sentence description. At this point, we had a Bag of Categories and a Bag of Visual Vectors to which we mapped the one-sentence description through the different pipelines.

To prevent additional complexity (like weighting), we decided to merge the outcome of the visual (*ResNet*) and text (category) regressions, and perform a K-Nearest-Neighbor search to find the 20 most relevant images in the combined outcome space.

Residual Neural Networks have not been as traditionally used as some of the other algorithms explored in this work. However, Lu and Qin have reported success in coupling a K-Nearest-Neighbor (KNN) algorithm with results from a Neural Network Model in the analysis of rainfall magnitude [15]. Drawing on their insights, we have decided to use KNN. Figure 8 demonstrates three distinct examples in which our analyses were particularly successful.

Since the distance metric (determining the similarity of two vectors in a space) is crucial to the performance of the KNN algorithm, we decided to explore few different options like the following: L2 Norm (Euclidean), L1 Norm, Cosine Similarity (normalized dot product) and Pearson Correlation Coefficient. We achieved highest performance with cosine similarity (score=0.48), which was anticipated because Euclidean distance is not good with handling data from different scales (Figure 7).

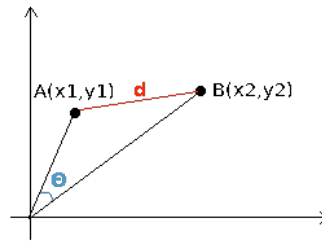


Figure 7: Showing the difference between cosine similarity (theta) and Euclidean distance (d) in a two dimensional space

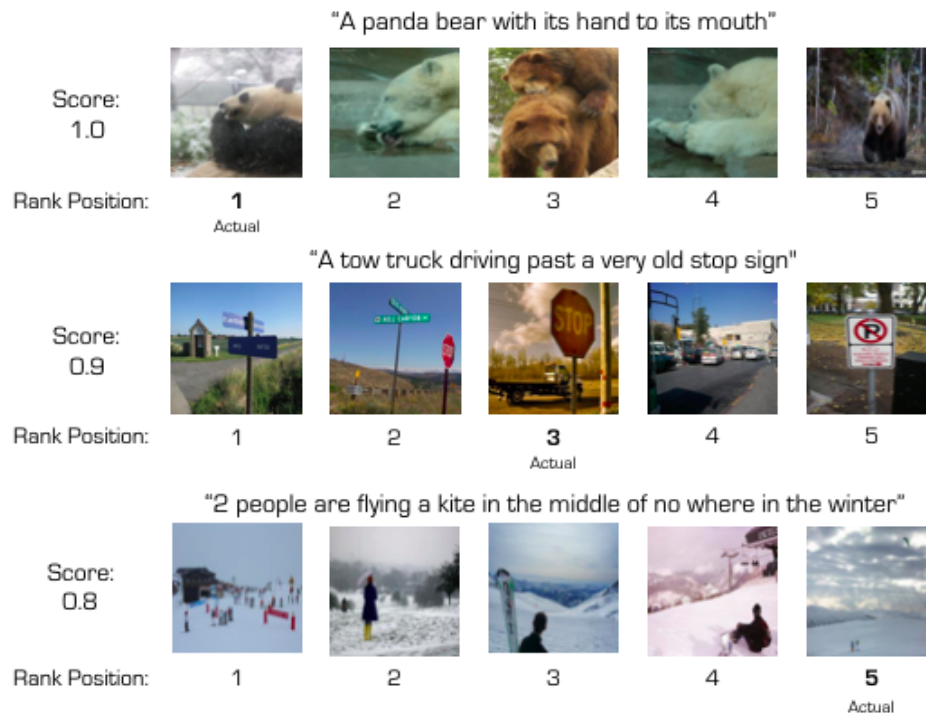


Figure 8: Three instances of algorithm matching resulted in 1.0, 0.9, and 0.8 scores, respectively

6 Results: The Combined Model

Using Cornell Tech's *Image Ranking Leaderboard* as an evaluation criterion, this algorithm received a score of 0.306 on Kaggle -- ranking it as the fourth most effective algorithm submitted. Considering the time constraints of this project, this result is proof of the immense value of combining text elements with RNN features in the rank-ordering of images.

To score and evaluate the performance of the framework, we used the Mean Average Precision at 20 (MAP@20) metric. For each sentence, there was only one true corresponding image. As we were looking to rate a list of top 20 search results, MAP@20 was an effective method to score our model in the range [0,1] (Figure 9) using the following intuition: if the corrected image was ranked first, the model got 1 point; if it was ranked 11th, the model got $\frac{1}{2}$ point. If the corresponding image was not in the top 20 list returned by the model, it got 0 points for that image (but not a negative number). The overall score was calculated as the average score on a 2,000 item long image set.

$$score = \frac{20 + 1 - i}{20}$$

Figure 9: Scoring with MAP@20

While observing our results, we found that our model was highly effective in predicting images that had large areas of a single texture or color. However, images where smaller details dominated the photo were classified with reduced accuracy levels.

While our results do not yield the same accuracy rates as those of models that have been developed over larger amounts of time, they show the promise of a combined analysis, which employs both text elements and visual features obtained from RNN.

7 Conclusion: Towards Comprehensive Image Processing

We live in a time in which Google Images and YouTube are competing with traditional Google search on number of hits and scale [16]. There is, therefore, an urgent need to understand how we could best employ the tools in our service to create meaningful and accurate analyses. We had started this series of experiments with the goal of running two disparate algorithms: one trained on text and the other on RNN-driven visual data. However, we found that separately, these algorithms provided us with much less accuracy than when combined. As the field of neural network analysis grows, machine learning researchers would be wise not to abandon text-driven analytics, but rather incorporate the two methods into a combined approach to image processing -- which might just be the future of this field.

Appendix

Processing Step	Approach	Score	Comments
Preprocessing	Bag of words without SVD	0.19	<ul style="list-style-type: none"> Curse of dimensionality
	ResNet features without SVD	0.19	
Mapping Description to ResNet	K Nearest Neighbor	0.27	<ul style="list-style-type: none"> Big feature vectors Works but inferior Works but not the best High sensitivity, Overfitting
	Linear Regression	0.39	
	Lasso	0.42	
	Decision Trees (bagged)	0.18	
Mapping Description to Tags	K Nearest Neighbor	0.35	<ul style="list-style-type: none"> Big feature vectors High sensitivity, Overfitting Works but not best Too complex, a black box
	Random Forest	0.12	
	SVM	0.45	
	Multilayer perceptron (NN)	0.15	
Individual pipelines	Tag analysis	0.27	<ul style="list-style-type: none"> Combination of models performed better
	RNN feature analysis	0.39	
Search Using KNN	Euclidean Distance	0.42	<ul style="list-style-type: none"> Works but not great Works but not the best
	Pearson correlation coefficient	0.45	
Ensemble of Weak Algorithms	Bagged KNN in search	0.02	<ul style="list-style-type: none"> Averaging ranks Averaging effect Not better than the individual max
	Boosting (Gradient boosting)	0.32	
	Voting of Log Reg and SVM	0.42	

Table 2: Training Data Scores via Cross Validation with 70-30 train-test split on different models

References

- [1] Google Operating System. (2013). How Google's Image Recognition Works [Blog post]. Retrieved from <http://googlesystem.blogspot.com/2013/06/how-googles-image-recognition-works.html>
- [2] Wright, J., Ma, Y., Maira, J., Sapiro, G., Huang, T., Yan, S. (2009). Sparse Representation For Computer Vision and Pattern Recognition. *Proceedings of IEEE*, 97(1), 1-10.
- [3] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.
- [4] Jing, Y., & Baluja, S. (2008). Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11), 1877-1890.
- [5] Bernard J. Jansen, (2008) Searching for digital images on the web, *Journal of Documentation*, Vol. 64 Iss: 1, pp.81 - 101
- [6] Greenberg, J. (1993), Intellectual control of visual archives: a comparison between the Art and Architecture Thesaurus and the Library of Congress Thesaurus for Graphic and Materials, *Cataloging and Classifications Quarterly*, Vol. 16 No. 1, pp. 85-101.
- [7] Gebraeel, N., Lawley, M., Liu, R., & Parmeshwaran, V. (2004). Residual life predictions from vibration-based degradation signals: a neural network approach. *IEEE Transactions on industrial electronics*, 51(3), 694-700.
- [8] Cui, J., Wen, F., & Tang, X. (2008). Real time google and live image search re-ranking. In *Proceedings of the 16th ACM international conference on Multimedia* (pp. 729-732).
- [9] Birkett, A. (2014). F-Patterns No More: How People View Google & Bing Search [Blog Post]. Retrieved from <http://conversionxl.com/how-people-view-search-results/>
- [8] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, 2005.
- [9] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Novel reranking methods for visual search. *IEEE Multimedia*, 2007
- [10] Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4), 43-52.
- [11] Ientilucci, E. J. (2003). Using the singular value decomposition. *Rochester Institute of Technology, Rochester, New York, United States, Technical Report*.
- [12] Yang, J., Jiang, Y. G., Hauptmann, A. G., & Ngo, C. W. (2007, September). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval* (pp. 197-206). ACM.
- [13] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55-67.
- [14] http://scikit-learn.org/stable/auto_examples/neural_networks/plot_mlp_alpha.html, sklearn regularization comparison
- [15] Lu, Y., & Qin, X. S. (2014). A coupled K-nearest-neighbour and Bayesian neural network model for daily rainfall downscaling. *International Journal of Climatology*, 34(11), 3221-3236.
- [16] Cheng, X., Dale, C., & Liu, J. (2008, June). Statistics and social network of youtube videos. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on* (pp. 229-238). IEEE.