

# A Multimodal Learning to Rank model for Web Pages



Nikhila T Bhuvan, M Sudheep Elayidom

**Abstract:** “Learning-to-rank” or LTR utilizes machine learning technologies to optimally combine many features to solve the problem of ranking. Web search is one of the prominent applications of LTR. To improve the ranking of webpages, multimodality based Learning to Rank model is proposed and implemented. Multimodality is the fusion or the process of integrating multiple unimodal representations into one compact representation. The main problem with the web search is that the links that appear on the top of the search list may be either irrelevant or less relevant to the user than the one appearing at a lower rank. Researches have proven that a multimodality based search would improve the rank list populated. The multiple modalities considered here are the text on a webpage as well as the images on a webpage. The textual features of the webpages are extracted from the LETOR dataset and the image features of the webpages are extracted from the images inside the webpages using the concept of transfer learning. VGG-16 model, pre-trained on ImageNet is used as the image feature extractor. The baseline model which is trained only using textual features is compared against the multimodal LTR. The multimodal LTR which integrates the visual and textual features shows an improvement of 10-15% in web search accuracy.

**Keywords:** Learning to Rank, LETOR, LTR, transfer learning.

## I. INTRODUCTION

Learning to rank algorithm is a machine learning algorithm that ranks the documents automatically using an extracted feature set [4]. Learning-to-rank algorithms optimally combine features exacted from query-document pairs through discriminative training. It even can be used for rank aggregation like in the case of metasearch engines. Learning to rank becomes very useful in the case of search engines as daily they get a huge lot of training data in the form of user feedbacks and search logs. This can constantly improve their ranking mechanism. The ranking problems in IR could be tackled using two major approaches: the learning to rank (LTR) approach and the traditional approach (non-learning) such as BM25, language models, etc. The LTR model automatically learns the parameters of the ranking function by training whereas other methods heuristically determines the

ranking function. The heuristic tuning will be feasible if the ranking model has only a few parameters. It becomes difficult for a non-learning approach to establish a ranking function incorporating all the feature values when the number of parameters is more. In contrast, multiple shreds of evidence can be of good use for the learning to rank approach. This paper focuses on supervised ranking on how to order the webpages of a web search query much more efficiently, using a ranked list of query document pairs as training and testing datasets. For each request  $q_k$ , there is an associated set of webpages  $\{W_{k1}, W_{k2}, \dots, W_{kn}\}$ . The main focus is on how to order these offerings in a user satisfying manner. In learning to rank method both query and webpages are represented as feature vectors or feature values. A query  $q$  and its associated webpage  $w$  can be represented by a feature vector  $x$ , where  $x = \Phi(w, q)$ .  $\Phi$  is a feature extractor function based on BM25 or PageRank or frequencies of query terms in the webpage. The image feature is extracted through a deep learning model using a transfer learning technique from a pre-trained VGG-16 model.

In earlier days, a probabilistic method was used to rank the documents, whereas now, it's an automatic process of learning based on the training data. The training data that is provided to the learning model will be the feature vector of a web page. This is multimodal learning to rank model that uses images and textual features to rank the webpages. The feature from images and webpages are used to train the model which is used to re-rank the webpages. The visual features extracted from the images of user interest and 46 webpage textual features are considered as the feature vector to train the model.

The work provides a relative ordering of webpages based of multiple modalities like text and images. The LETOR dataset is the collection of textual features of the webpages. The LETOR dataset is extended to add the image features of user interest extracting the image features using transfer learning. This extended feature vector is used to rank the webpages to provide a better Mean Average precision than the existing ranking algorithms like Ranknet, Adarank, etc, that are already implemented in the LETOR[5] dataset.

The major contribution of the work to the research community can be stated as:

- Proposal to enhance the LETOR dataset using image features of a webpage in improving research in this area.
- Use of transfer learning to extract features of out of domain images and use these features along with the textual features to rank the web pages using deep neural networks.

Revised Manuscript Received on August 06, 2020.

\* Correspondence Author

**Nikhila T Bhuvan\***, Department of Information Technology, Rajagiri School of Engineering & Technology, Rajagiri Valley, Kakkanad, Kochi, Kerala, India. E-mail: nikhilatb@rajagiritech.edu.in

**M Sudheep Elayidom**, Division of Computer Science, School of Engineering, CUSAT, Kerala, India. E-mail: sudheep@cusat.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

### II. RELATED WORK

Multimodal fusion is one of the trending topics in the multimodal research area. Artificial intelligence has paved way for this promising field with the help of machine learning and Deep learning networks.

The fusion of multimodal data is integrating the features from two different domain and understanding the influence of one over the other. The design goal of such fusion models is to combine the multiple modalities to make use of this heterogeneous data to complement each other and provide more robust predictions. The complexity such models is the challenging factors. Fusing multiple modalities, while still maintaining reasonable model complexity is a challenging task. Multimodal fusion is one of the trending topics in the Multimodal research area. Artificial intelligence has paved way for this promising field with the help of machine learning and deep learning networks. The fusion of multimodal data is integrating the features from two different domains and understanding the influence of one over the other. The design goal of such fusion models is to combine the multiple modalities to make use of this heterogeneous data to complement each other and provide more robust predictions. The complexity of such models is challenging factors. Fusing multiple modalities, while still maintaining reasonable model complexity is a challenging task. Multimodality search ranking could be traced back to time 2008 when Zau et.al[7] used image snippets extracted from Web pages along with the text snippets to the user for result presentation. A relevance feedback mechanism was also presented along, to the user for image suggestion and further improvement of the entire system. Nowadays, instead of the explicit feedback from the user, the implicit feedback is taken from the user using artificial intelligence. The image snippet of the webpage was more convenient for the user to identify the Web pages that the user expects. [6] A multimodal web reranking that combines the text ranking with the image descriptors exploited a pure text-based search engine to find a ranked list of candidate web pages for a given query. Subsequently, the same query is used to collect the images related to it from its image search. The visual descriptors are extracted using the classifiers and the candidate set of webpages is reranked using visual information extracted from the images contained with the page's textual content.

Learning to rank algorithms captures most of the relevance signals in a Web page and a large number of features are usually generated. Features generated are computed based on the extracted term frequency, idf and lot more textual elements along with link analysis, and user logs. Multimodalities like webpage layout, images, heat maps, and videos on the webpage can be used to provide a better rank for the webpages. They [1]approached Web pages not as a set of linked texts, but as the one that has a structured layout organizing a large variety of elements in different styles. They implemented a methodology taking advantage of such layout that can convey useful visual information, indicating the relevance of a Web page. For example, the Both query-independent layout like raw page layout and query-dependent layout are taken into consideration. This rich structural information tells a lot about the page relevancies. However, Web search in the past seldom explored such visual information of layout. The rich query independent visual features were learned automatically from

the layout of Web pages (i.e., Web page snapshots) for relevance ranking. Both query-independent and query-dependent snapshots are considered as the new inputs in this visual perception model.

DeepRank [2], another deep learning approach, was proposed to generate relevance ranking of webpages without explicit understandings of the relevance. A deep learning architecture to simulate the human judgment process of judging the relevancy is implemented. They designed a detection strategy to extract relevant contexts. Then, a measuring network is utilizing a Convolutional Neural Network (CNN) is used to determine the local relevance. Finally, these local relevancies are sequentially integrated to produce a global relevance score. IR characteristics including exact/semantic matching signals, proximity heuristics, query term importance, and diverse relevance requirement are well captured by the model. DeepRank could outperform much learning to ranking methods. This was proved through their experiments on benchmark LETOR dataset and a large scale click-through data. The performance of learning to rank (LTR) can be used to improve using the visual appearance of a webpage. They carry valuable information about the page's quality that helps in improving the performance of LTR [3]. The visual features extraction methods that utilize transfer learning from a pre-trained image classification model and synthetic saliency heat maps generated from webpage snapshots are integrated to form the Visual learning TO Rank (ViTOR) model. The ViTOR model is not a multimodal LTR but they used a transfer learning concept to generate the dataset that consists of visual snapshots, nonvisual features, and relevance judgments for ClueWeb12 web pages and TREC Web Track queries. The ViTOR model significantly improved the performance of LTR with visual features. Their work was inspired by human visual search behaviours on page viewing to extract the visual features. Another work that closely resembles the methodology accepted by us is being done and published by Etsy[17], an American e-commerce website. They used the concept of multimodality based search ranking using deep Convolutional networks to improve the quality of search results. The methodology proposes a multimodality search and explains how image information can be successfully used to separate pairs of highly different items that are ranked similarly by a text-only model. Complementary information in the images that is unavailable in the text could be explored to provide better search results. The high level of visual information in the images are extracted using deep Convolutional Network using the concept of transfer learning. The pre-trained VGG-19 is used to extract the feature vector of the image and the feature vector of the text using Bag of Words method is also extracted to form a combined feature vector. This is matched with the feature vector of the search query to provide the rank list of the images to be listed to the user. The role of learning to rank is to learn from the historical data. Once the sorted list is displayed to the user, the user clicks on the images are learned to fine filter the list in the upcoming searches. Incorporating extracted visual features from the images along with the textual description can provide better accuracy in the ranking of web pages, than the single modality ranking.

Since the textual features are normally noisy, the text-only ranking model is insufficient for the web search. So, we propose to jointly utilize the textual and visual features to boost the learning ranking model. Most of the related papers discussed explored the visual layout of the web pages, whereas we are trying to utilize the actual pictorial contents on the webpage along with its textual content to rank the web pages.

### III. MULTIMODAL LTR: ARCHITECTURE

There are mainly 3 main approaches in learning to rank pointwise, pairwise, and listwise approaches [4]. In the pointwise approach, the relevancy of a document is predicted. It ignores the group-wise classification of documents in terms of multiple queries. In the pairwise approach, training data is organized as a query and its associated documents along with its relevancy score. The classifier classifies the ranking order of the document pair related to a query. The rank lists of a query are taken as instances in the listwise learning approach. The ranking model  $f(x)$  is learned from the training data which is the query and its associated rank list along with their relevancy score. Here, the supervised pointwise approach is implemented. Each query has got its own 46-dimensional textual features associated with its relevancy score. This is used to train the model at first. Then another model is trained using these 46 features along with the 32 features of its associated image which is extracted [13] using transfer learning as in the figure given. The relevancy score provided by the second model is about 10-15% better than the model that was trained using just the textual features.

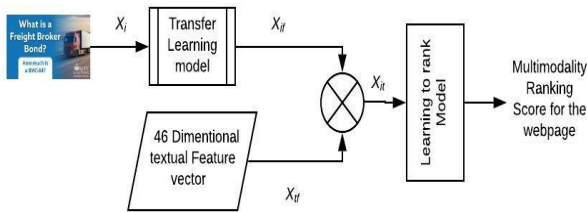


Fig.1. Architecture of multimodal LTR

The architecture of the multimodal model is visualized in Fig 1. Images,  $X_i$  are fed into the transfer learning model to extract its visual feature. The feature extraction is done using VGG-16[9] model. The first feature extraction layers create a generic visual feature vector as they use Convolutional filters trained on the ImageNet dataset[10]. These generic features are transferred to image specific by adding more fully connected layers trained on the Images specific to the queries and generates the feature vector,  $X_{if}$ . This is combined with the textual feature vector of the webpages,  $X_{if}$  to train the LTR model.

### IV. DATASET

#### A. Textual Features

LETOR4.0[5] has the benchmark collection of the feature set of documents/URLs which can be used for research on Information Retrieval released by Microsoft Research Asia. LETOR is a collection of indexed document corpora, both normal text documents, and web pages and its corresponding

queries, which are classified as training, validation and test set. Since its release LETOR is very widely used by the research community. The documents related to queries are represented as 46 dimensionality feature vectors extracted for each document. Each query is represented using a query ID, which can be mapped using the TREC million dataset. A sample screenshot of LETOR is given below as Fig.2.

```
2 qid:10032 1:0.056537 2:0.000000 3:0.666667 4:1.000000 5:0.067138 ...
45:0.000000 46:0.076923 #docid = GX029-35-5894638 inc =
0.0119881192468859 prob = 0.139842
0 qid:10032 1:0.279152 2:0.000000 3:0.000000 4:0.000000 5:0.279152 ...
45:0.250000 46:1.000000 #docid = GX030-77-6315042 inc = 1 prob = 0.341364
0 qid:10032 1:0.130742 2:0.000000 3:0.333333 4:0.000000 5:0.134276 ...
45:0.750000 46:1.000000 #docid = GX140-98-13566007 inc = 1 prob =
0.0701303
1 qid:10032 1:0.593640 2:1.000000 3:0.000000 4:0.000000 5:0.600707 ...
45:0.500000 46:0.000000 #docid = GX256-43-0740276 inc =
0.0136292023050293 prob = 0.400738
```

Fig 2. Screen shot of Supervised dataset of LETOR 4

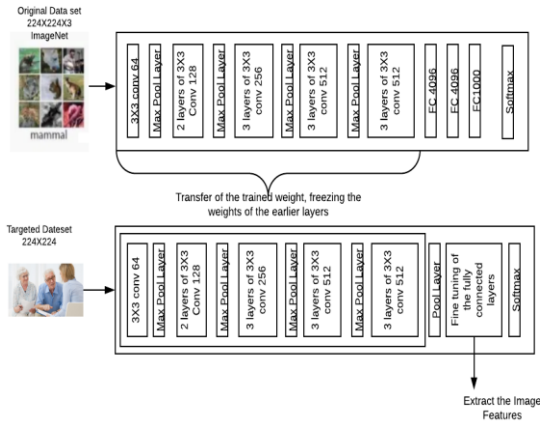
LETOR 4.0 has four ranking settings, supervised ranking set, Semi-supervised ranking set, Rank aggregation, and List-wise ranking set. The supervised ranking set is used to train and test the model. Each row represents a query-document pair. The relevance label of each pair forms the first column. The relevance labels are either 0 or 1 or 2. More value of the relevance label shows that the query document is more relevant. A '0' value indicates that they are not related to each other. The second column is the query id, followed by the features and its values associated with each query document pair features, and the last column of each row is the probability factor including id of the document. A 46-dimensional feature vector is used to represent a query-document pair. Currently, there is no combined dataset for research on the multimodality dataset. All the datasets that are available for Learning to Rank like Yahoo! LETOR dataset, Yandex imat'2009 dataset, Istella Learning to Rank dataset etc are the one which has the textual features of the webpages. Our work can even be considered as a contribution towards the creation of a multimodality dataset for Learning to Rank that could provide a significant improvement in the LTR performance. From the literature, it was found that these webpages could not be useful for the research on visual features as it does not have any visually rich webpages as today. Thus, reached a conclusion on using the textual features from the LETOR dataset and generate the visual features from images related to the query used to generate the GOV2 dataset. The queries were taken from TREC web tracks, Million Query 2008, and Million Query 2007. The next section explains the feature vector generation for images.

#### B. Image feature set generation using transfer learning

Transfer learning[8] could be defined as a machine learning methodology by which a model developed for achieving a task is exploited for another related task. Transfer learning[16] can also be thought of as an optimization methodology for domain adaptation. In transfer learning, a base network is trained on a general dataset, and this model is repurposed to utilize the learned features or transfer them, to a second target network.

One advantage of transfer learning is that it is an optimization technique, a shortcut to save time or get better performance. Another advantage is when there is a lack of training data. When the scarcity of data is there, these trained models can be utilized to build customized models.

Examples of such models will include Oxford's VGG model, Google's Inception model, Microsoft's ResNet Model, DenseNet and NASNet are the most modern pertained models in this series.



**Fig.3: Using transfer learning to extract features**

All deep learning architecture follows a layered model that learns different features at different layers. These layers are finally connected to the output layer. The different layers in a CNN model are:

- Convolutional layers (CONV): These layers perform the feature extraction of images. A CONV layer using the filters and strides is used to detect specific features in the input image.
- Activation layers (ACT): This layer is used to implement functions that help to decide if the neuron to be activated. The activation functions that are generally used in CNN are Sigmoid, Tanh, and Rectified Linear Unit (ReLU).
- Pooling layers (POOL): To control the overfitting of the model some subsampling is required on the number of parameters extracted which is done by the pooling layer.
- Fully connected layers (FC): FC layer uses all the activations from its previous layers to classify the input by connecting all the activated neurons from the previous layer. The FC layer takes as input a huge dimension and outputs an N dimension which is equivalent to the number of classes.
- Classification layers (CLASS) The classification layer is the one that performs the final classification that is usually implemented using a sigmoid activation function for a binary class problem or using a SoftMax function with multiclass logistic regression problems.

To use images to train the LTR model, the images need to be converted to the feature vector. The feature extraction from an input image can be achieved by the elimination of the final output layer(Softmax layer) of any deep pre-trained model[11]. This process is less costly than training a new model. VGG-16 model is a well-explored image classification model that is pre-trained on ImageNet parameters. It is commonly used for transfer learning the features. Its architecture takes an image of size 224X224 as input. VGG consists of a set of convolutional layers and two fully connected layers with 4096 neurons in each layer. The

features extracted by the convolutional layers [11]are used by the fully connected layers to classify it using the final Softmax layer that has 1000 neurons to classify the 1000 classes of the ImageNet. These convolutional layers that extract features from an input image, can be used to extract the generic visual features of any image. All 4096 feature vector of the image cannot be taken to concatenate with the 46 textual features as it will anyways result in overfitting. The fully connected layers of VGG-16, instead, can be altered and retrained to be used with new inputs and tasks. So, four new layers are introduced, to extract the features, and to reduce the dimensionality of the feature vector to 32. The first fully connected layer has 4096 neurons which is reduced to 1024 in the next layer which in turn gets reduced to 128 and finally in the last layer we have 32 features. Following the fully connected layer we have the softmax layer using which the new images are classified. A training accuracy of 97% and the validation accuracy of 90% was achieved for the model. The new layer is trained using the images specific to the queries in MQ2007 and MQ2008. The LETOR dataset is also based on MQ2007 and MQ2008[12] query set of TREC web tracks[14]. The parameters of the fully connected layers of VGG-16 is optimized during training. The Softmax layer is removed from the model to extract the feature vector for the images. The feature dimensionality of Xif is set to 32.Fig.3 explains each step of it.

### V. EXPERIMENTAL SETUP

The experiment is done using the supervised ranking set of LETOR.4. Initial preprocessing is done on this ranking set to extract the query document pair that has a minimum count of at least 80 samples. There were around 40 query document pairs with an average sample size of 120 webpages. The baseline architecture is the one that is trained only using textual features without adding the visual features. This baseline architecture is compared against the multimodality LTR architecture trained using both visual features extracted using VGG and textual features from LETOR. Below given is the sample table of queries of TREC web track and the number of training samples on which the model was trained on. These QIDs matches the QIDs in the LETOR dataset.

The images for these queries are scrapped to form the internet using SerpApi forming a new dataset that is used to train the fully connected layers of the VGG-16 model. SerpApi would scrape, extract, and make sense of the information displayed through the search engine algorithm. The retrained VGG-16 model extracts the 32 dimension feature vector of images. These 32 image features added along with the 46 textual features form the new feature vector on which the LTR is trained. 5-fold cross-validation is done using the dataset.

**Table 1. Sample Query set along with the QID and the number of samples available for training**

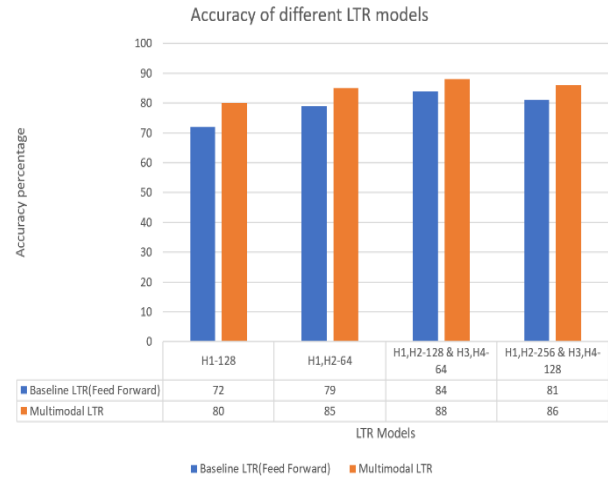
QID	No.of samples	Query
2297	147	official state of california web site
12793	121	pictures of presidents
12904	119	lyrics you are my sunshine my only sunshine
15526	119	distance from phoenix to san diego
13782	118	hurricane katrina in pictures
10078	118	attorney for senior
10419	118	when a hog is ready to farrow
11759	118	what are the nj state departments
11893	118	what is respiratory therapy
14910	117	how to pass a blood pressure test
15903	117	periodic table of the elements photo
12385	116	how do you know if you have aspergillus
13194	116	goodwill industries orlando florida
13376	116	nasa marshall space flight center huntsville ala
13499	115	what is a freight broker bond
12165	115	district of columbia department of motor vehicles
11565	114	dr. geraldine chapey
10680	113	homeless shelters in new york city

**VI. RESULTS**

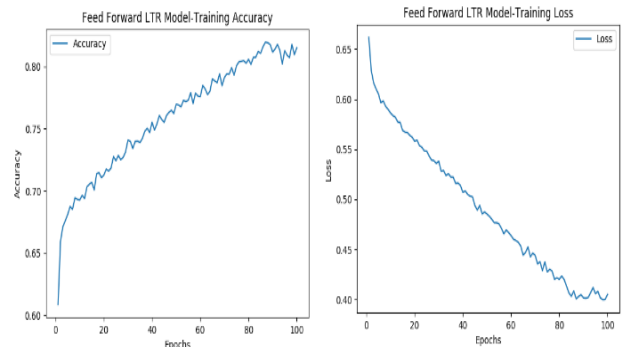
Many sequential models with one hidden layer with 128 neurons, 2 hidden layers with 64 neurons each were implemented. Two models with 4 hidden layers were also implemented with layers one and two having 128 and 256 each and layer three and four with 64 and 265 each were also tried out for relevancy prediction. From the table, it is clear that the model with four hidden layers with 128 neurons in the 1st and 2nd hidden layer and 64 neurons to the 3rd and 4th hidden layer provides the best accuracy. The baseline model was trained on 100 epochs and the combined multimodal model was trained on 30 epochs. The loss accuracy graph of both is shown in Fig.4 and Fig 5. The baseline model used the feed-forward model for its implementation. The below given Table.2 summarizes the validation accuracy of the baseline LTR model and the multimodality LTR model.

**Table 2. validation accuracy of the LTR model**

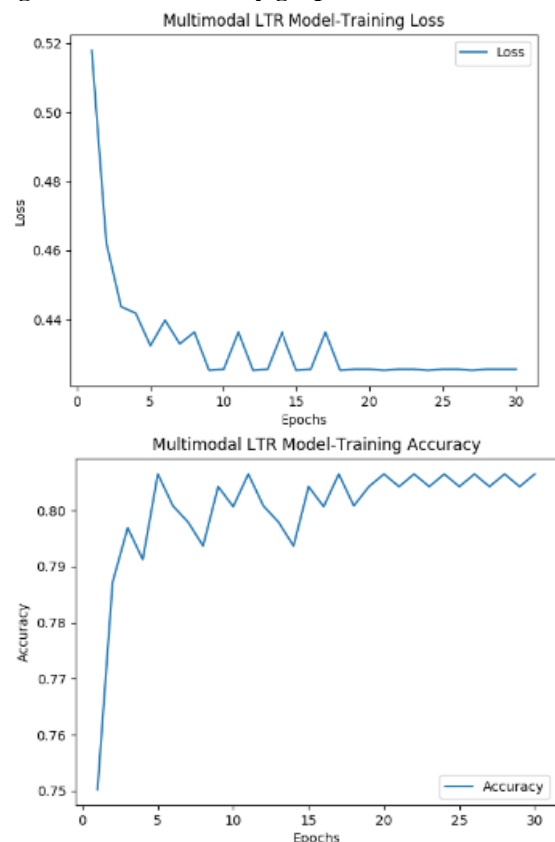
LTR Model	No.of hidden layers and neurons in each layer	Baseline Accuracy (Feed forward Model)	Multimodal Accuracy (combining with transfer learning model)
3 layer model	1 hidden layer H1-128	72%	80%
4 layer model	2 hidden layer H1-64 H2-64	79%	85%
6 layer model	4 hidden layer H1-128 H2-128 H3-64 H4-64	84%	88%
6 layer model	4 hidden layer H1-512 H2-512 H3-256 H4-256	81%	86%



**Fig.3.Graph Showing Accuracy of Different models**



**Fig.4.The loss -accuracy graph of the Baseline Model**



**Fig.5.The loss -accuracy graph of the Multimodal Model**

The observations made can be summarized as:

- Adding dropout and L2 regularization layers decreased the overfitting of the model.
- ReLu is the activation function used.
- The loss function used is cross entropy categorical loss function.
- Learning rate is not kept constant, instead, Adadelata is used as the learning rate optimizer.
- The final layer is Softmax layer.

### VII. CONCLUSION AND FUTURE ENHANCEMENT

The implementation clearly shows that multimodality features provide an improvement of 10-15% percentage in search accuracy. The LETOR dataset now has only textual features, it could be further extended to include the image feature set and video feature set into it. These would give more scope in the research field of LTR. Another direction for future work is the implementation of multimodality based LTR on state of art ranking models like such as RankBoost and LambdaMart to learn the influence of multiple modalities on LTR. There are lots of researches going around in the field of heat maps, visual features extracted from vanilla snapshots, etc along with the LETOR dataset which is showing an improvement in the webpage ranking. As a future research topic, the combination of these multiple visual features like heat maps, visual features extracted from vanilla snapshots, spacial features of image and webpages can be considered to provide promising results.

### REFERENCES

1. Yixing Fany, Jiafeng Guo, Yanyan Lan, Jun Xuz, Liang Pang, Xueqi Cheng, "Learning Visual Features from Snapshots for Web Search", CIKM'17, ACM, November 6-10, 2017.
2. Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, Xueqi Cheng, "DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval", arXiv:1710.05649v2 [cs.IR] 22 Jul 2019
3. Bram van den Akker , Ilya Markov, Maarten de Rijke," ViTOR: Learning to Rank Webpages Based on Visual Feature", IW3C2 (International World Wide Web Conference Committee, May 13-17, 2019, San Francisco, CA, USA
4. Hang, Li, *Learning to rank for information retrieval and natural language processing. Synthesis Lectures on Human Language Technologies*, Morgan & Claypool, 2011.
5. Tao Qin ,Tie-Yan Liu," Introducing LETOR 4.0 Datasets", arXiv preprint arXiv:1306.2597 (2013)
6. Sergio Rodriguez-Vaamonde, Lorenzo Torresani, Andrew W. Fitzgibbon, "What Can Pictures Tell Us About Web Pages? Improving Document Search using Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Oct 2014.
7. Z.-H. Zhou and X.-B. Xue, "Improving Web Search Using Image Snippets", ACM Transactions on Internet Technology (TOIT), Volume 8, Issue 4, September 2008.
8. Dipanjan (DJ) Sarkar Nov 15, 2018 [online] Available: <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>.
9. Karen Simonyan and Andrew Zisserman,"Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556 (2014).
10. Alex Krizhevsky, Ilya Sutskever, and Georey E Hinton., "Imagenet classification with deep convolutional neural networks", In NIPS. 1097-1105(2012)
11. Eric C. Orenstein , Oscar Beijbom," Transfer learning and deep feature extraction for planktonic image dataset", IEEE Winter Conference on Applications of Computer Vision, 2017.
12. Ben Carterette, Virgil Pavlu, Hui Fang, Evangelos Kanoula, "Million Query Track 2009 Overview", NIST Special Publication
13. Jeff Donahue , Yangqing Jia , Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell, "DeCAF: A Deep Convolutional

- Activation Feature for Generic Visual Recognition", Proceedings of the 31st International Conference on Machine Learning, PMLR 32(1):647-655, 2014.
14. TREC. TREC 2009 Million Query Track - Prels relevance judgements. Website, [online] Available:<http://trec.nist.gov/data/millionquery/09/prels.20001-60000.gz>.
15. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze,"An Introduction to Information Retrieval", Cambridge University Press Cambridge, England, Printed on April 1, 2009
16. Fchollet April 2020 [online] Available : <https://keras.io/getting-started/sequential-model-guide/>
17. Corey Lynch, Kamelia Aryafar, Josh Attenberg,"Images Don't Lie: Transferring Deep Visual Semantic Features to Large-Scale Multimodal Learning to Rank", KDD '16, August 13-17, 2016, San Francisco, CA, USA

### AUTHORS PROFILE



**Ms. Nikhila T Bhuvan** is working as an Assistant professor in the Department of Information Technology at Rajagiri School of Engineering & Technology, Kakanad, India. Currently a part time doctoral student in School of Engineering, Cochin University of Science and Technology(CUSAT), Kochi, India. She completed her BTech from the Kannur University and MTech from IGNOU. She is the chapter author for the book, "Data Mining and Warehousing". Her areas of Interest are Cloud Computing, Information Retrieval, and Machine Learning.



**Dr. M. Sudheep Elayidom** is a computer engineer with a PhD in data mining. Currently he is working as a Professor and PhD guide at School of Engineering, CUSAT, Kochi, India. He completed his B.Tech from the MG University and M.Tech from CUSAT, with first ranks in both. He has several international publications to his credit. His areas of interest include data mining, databases, software engineering, object oriented analysis, Big Data, Cloud databases, etc. He is the author of the book by the name "Data Mining and Warehousing", which is a syllabus prescribed text book for courses on Data Mining in many universities across the country.