Ensemble Learning on Deep Neural Networks for Image Caption Generation

Harshitha Katpally
Arizona State University
School of Computing, Informatics, and Decision Systems
Engineering, Mesa AZ 85212
hkatpall@asu.edu

Ajay Bansal
Arizona State University
School of Computing, Informatics, and Decision Systems
Engineering, Mesa AZ 85212
ajay.bansal@asu.edu

***Abstract*—Capturing the information in an image into a natural language sentence is considered a difficult problem to be solved by computers. Image captioning involves not just detecting objects from images but understanding the interactions between the objects to be translated into relevant captions. So, expertise in the field of computer vision paired with natural language processing is crucial for this purpose. The sequence to sequence modelling strategy of deep neural networks is the traditional approach to generate a sequential list of words which are combined to represent the image. But these models suffer from the problem of high variance by not being able to generalize well on the training data. The main focus of this paper is to reduce the variance factor that will help in generating better captions. To achieve this, Ensemble Learning techniques have been explored, which have the reputation of solving the high variance problem that occurs in machine learning algorithms. Three different ensemble techniques namely, k-fold ensemble, bootstrap aggregation ensemble and boosting ensemble have been evaluated in our work. For each of these techniques, three output combination approaches have been analyzed. Extensive experiments have been conducted on the Flickr8k dataset which has a collection of 8000 images and 5 different captions for every image. The bleu score performance metric, which is considered to be the standard for evaluating natural language processing (NLP) problems, is used to evaluate the predictions. Based on this metric, the analysis shows that ensemble learning performs significantly better and generates more meaningful captions compared to any of the individual models used.**

***Keywords*—deep neural networks, image captioning, ensemble learning, bootstrap aggregation, boosting, k-fold ensemble.**

I. Introduction

The emergence of the field of deep learning has helped in solving an enormous number of problems for some time now. One among these problems, is the problem of automatic image caption generation by computers, which has many applications in different domains. Some of the applications of image captioning include helping visually impaired people understand what the images contain by providing short descriptions, communicate to clinical experts regarding potential disease conditions found in medical images and to convert images to text, which can be used in certain applications where inferences are made only from textual data. With all the advantages of image captioning mentioned, it is crucial to develop state-of-the-art algorithms that can solve this problem and generate captions with acceptable accuracy.

As the problem deals with image processing, it can be classified as a computer vision problem, but including an additional step of generating textual sentences. So, a solution to this problem will have two phases, 1. A feature extraction phase, where features present in the image will be extracted and represented in the form of a feature vector, 2. A sentence generation phase, which takes as input the feature vector and a sequence of words are generated one after another which when combined together gives a meaningful description of the image. While the fundamental approach to solving this problem is established, there is still scope for research in this area, towards designing new models that can improve the accuracy of predictions. The main focus of this paper is to explore the effects of ensemble learning techniques on the problem of image captioning, which have long been proven to be very efficient in making better predictions when compared to single model settings [20]. The main challenge in image captioning is that of high variance, which occurs with the use of deep learning models as the models try to learn the specifics of training data. This means that the models used to solve image captioning have a lower tendency of generalizing on features from the training dataset and hence cannot give good predictions on new data. In this work, we identify the problem of high variance in image captioning and proposed an approach to solve it.

Though there are already existing sequence-to-sequence deep learning models for this purpose, we propose the use of ensemble learning techniques on these models. Deep learning neural networks can learn nonlinear complex relations in the data because of their deep structure and very large number of weights which get modified to represent these relations. But this nonlinearity gets reflected in the fact that the models tend to have a high variance, i.e. overfitting on the training data and not being able to generalize very well. One solution to reduce variance in deep models is to use the benefits of ensemble learning techniques which have been proven to solve the same. These ensemble techniques explore the predictions of different good models and combine them to get one single better prediction. Not just implementing ensemble learning on the problem but answering why and how the use of these techniques improves the performance over using a single model is the main goal. Our work explores various ensemble techniques that are available and analyzes the best ones suitable for the image captioning problem.

II. Related Work

There has been significant amount of research in generating captions for images using deep learning algorithms [5, 8, 18]. A series of CNN architectures have been designed to solve the image classification problem where objects in the images are classified into different classes. These CNNs are pre-trained on large image datasets which can be used directly without any further training for extracting features of images. But since these models are trained for classification, the outputs from the last second layer are used as feature vectors because the last layer gives a classification output, which is not necessary for this problem. Using these pre-trained models to extract image representations significantly reduces training time and are thus used widely. Several modifications have also been proposed to the sentence generation phase. Using the LSTM model with a copying mechanism for describing novel objects in the captions is one approach

978-1-7281-6332-1/20/$31.00 ©2020 IEEE
DOI 10.1109/ICSC.2020.00016

which helps in selecting words from novel objects at correct places in the sentence [1]. Another approach used to boost the prediction accuracy is using high-level image attributes in addition to the usual image representations. The relations between these attributes and image representations are explored to generate better captions. Attention mechanism is another important technique where different regions of the image can be weighed differently and depending on these weights, captions are generated [3]. This section gives a detailed description of each of the works mentioned above.

A. Feature Extraction Models

CNN is currently the state-of-the-art architecture for solving visual recognition problems. The core problem solved by them is the classification problem where objects in images are classified according to their class. A very widely used dataset for solving computer vision problems is the ImageNet dataset containing over 14 million images and all of the pre-trained networks are trained on the same dataset to classify as many as 1000 objects. This section discusses three of the architectures that have been proposed and considered to be the best at what they do. First is the VGG16 model proposed in [5] which achieves 92.7% top-5 test accuracy on ImageNet. There are two major drawbacks in the VGG network i.e. slow training process and the large values of the network weights which slow down the process are pertained in the network.

InceptionV3 is another model as proposed in [8] which has been proven to achieve an error rate of 4.2% which is less than the previously developed VGG networks. However, an advantage of InceptionV3 is that it performs a concatenation of multiple convolution outputs to incorporate a more precise representation of features in the network. This network also has a lower computational cost as compared to VGG which allows it to be used to work with large amounts of data.

The most recently developed pre-trained model is the ResNet50 model which allows for even better learning in deep networks when compared to InceptionV3 and VGG. Similar to both these architectures, a ResNet50 also is comprised of a series of convolution layers followed by fully-connected layers. But it is different from them in a way that it has a special component called residual networks in the architecture. With architectures like VGG, it would result in a problem of vanishing gradients if the network is just extended in the number of layers because, the deeper the network the lesser the chance for gradients to get updated. So, this results in a vanishing gradient problem as the networks get deeper. To resolve this problem, a residual network has been introduced and used in ResNet50. A residual network typically stacks multiple layers into residual blocks and applies an identity function so that the gradient is preserved [18]. This way multiple layers can be stacked together so the images can be trained on much deeper networks.

B. Sentence Generation Architectures

Along with feature extraction, research has also been done in the way sentences can be generated from the extracted features. A copy mechanism has been proposed in [1] where the architecture of the model is designed in a way to detect the novel objects in an image. The detected objects are then directly introduced in the generated sentence. A copy layer is introduced on top of the entire architecture to combine the LSTM network with the copying mechanism so that novel objects detected from copying can be directly included in the sentence generated by the LSTM. This approach towards generating captions helps in accommodating words in the sentence which closely relate to the objects in the images.

Another model produces image captioning with attributes [10] using a series of variant CNN and RNN architectures constructed by feeding the image representation along with attributes to the network in different ways to explore the relationships between them. Here, attributes are the properties seen in images which highly contribute to the salient objects in them.

Another very important model that has been proposed to solve image captioning with greater accuracy is the use of attention mechanism in the model as proposed in [11]. The reason that humans can generate captions with the greatest accuracy is because their brains work by giving attention to the important things and less to the ones that are not very important. This way it can capture the important aspects of what the eyes see and thus will be able to generate sentences by focusing more on the important aspects. This capability when introduced into a machine is known as attention mechanism. The encoder-decoder models often tend to not perform very well as the length of the sequence increases. Attention helps in addressing this limitation of handling long sequences and also in speeding up the learning process. This paper addresses the problem of high variance in deep learning models with the use of various ensemble learning methods for image captioning.

III. Ensemble Learning on Image Captioning

In this section we present our approach using ensemble learning methods for image captioning. Flickr8K dataset has been used in our work as it is relatively small compared to other available datasets and hence computations can be done faster. But it is also realistic and includes a good set of samples sufficient to learn good patterns from them. The dataset is available as open source. This dataset has a total of 8000 images with 6000 for training, 1000 for development and 1000 for testing. Every image in the dataset is associated with five different captions all of which can be used for training the model. One folder contains all of the 8000 images with unique identifiers as file names. Another folder contains four text files, where three of them have identifiers of the train, development, and test images, and the fourth file has image identifiers associated with all five different captions for every image in the dataset.

Image captioning system workflow comprises of the following steps" (i) *Feature Extraction*: images and image descriptions have to be pre-processed and prepared for training the model. Images have to be converted into vector representations that extract the critical features from images and represent them as float values. The size of the vector depends on the output layer size of the model being used for the purpose. (ii) *Cleaning Description Text*: involves preparing a vocabulary and preprocess it from the set of descriptions in the dataset and stores in a separate file. (iii) *Define the Model*: define the structure of the fundamental model that is used to generate captions as a sequence of words those when combined form a meaningful sentence explaining the input image. The caption generation phase is carried out by a recurrent neural network which generates words one after another. To train the model, a list of previously generated words has to be provided as input, and the next word is generated at the output. Converting the word representations into dense vectors provides capabilities to compare the correlations among words. After a word embedding is generated for all words in the vocabulary, the output from this layer is input the recurrent neural network layer [17]. This layer can be a simple RNN layer, a Long Short-Term Memory (LSTM) or a Gated Recurrent Unit (GRU) layer. All three networks have been explored in our work and used to generate a distinct set of ensembles. (iv) *Model Evaluation*: a trained model is then evaluated on a test dataset using an evaluation metric to calculate performance of the model. A metric that is most widely used for evaluating image captioning is the Bleu score (Bilingual evaluation understudy) [16]. Bleu scores range from 0.0 to 1.0 where 1.0 represents a perfect match and a 0.0 represents a perfect mismatch. Bleu score works by calculating the n-grams in a predicted sentence against n-grams in the reference sentences. The higher the match count, the higher will be the bleu score. N-grams are the number of words that are chosen for comparison each time.

Ensemble learning is the practice of training multiple good but different models instead of a single model and finally combining the predictions of all the models in a way suitable to the problem definition. This way, the predictions from ensemble learning can be better than the predictions from a single model itself [20]. In Figure 1, ensemble learning technique is applied on a classification problem, i.e. multiple classifiers are trained on the entire dataset or subsets of the dataset. A combined prediction of these classifiers is given as the final output which will be better than the output from a single classifier.

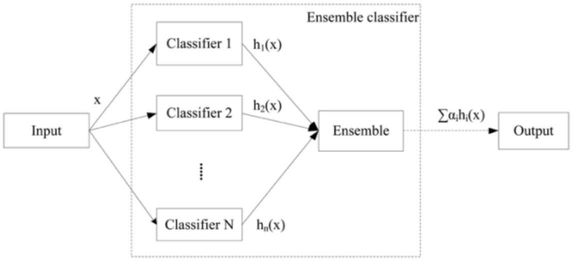


Fig 1. Basic Structure of Applying Ensemble Learning to a Classification Problem

Based on the different types of ensemble learning techniques we have chosen the following to experiment with, on the image captioning problem.

A. K-fold Cross Validation Ensemble:

This type of ensemble is generated when using different sets of training data to train different members of the ensemble. K-fold cross validation is the process used to understand how the machine learning algorithm responds on new data. The entire dataset is split into k-folds where k can be any integer value. If k is chosen to be 5, then the entire dataset is divided into 5-folds. Now, one of the folds is chosen to be a hold-out or test-dataset and all the remaining folds are considered as training data. The model is trained on this data and tested on the holdout dataset. This process is repeated for k-times where each fold gets a chance to act as a hold-out dataset. This approach can be used to generate an ensemble of models to achieve better results.

B. Bootstrap Aggregation Ensemble:

Bootstrap aggregation also comes under the category of ensemble learning by varying the training data size or composition for training different models. In this technique, a subset of the entire training dataset is chosen with replacement to train the model network. This approach is beneficial because it allows the models to expect a different density of samples in the training dataset when they are trained so they can reduce the generalization error. Implementing the bootstrap aggregation technique on the image captioning dataset has been pretty straight forward. Different number of samples have been generated from the training dataset with replacement and these samples are used to train different models based on the sample size. Finally, the predictions from each of the models have been combined to generate a final prediction.

C. Hyperparameter Tuning:

This is not a separate ensemble technique but a way to generate various models that can be used with varying training data or whose predictions can be combined in different ways for better results. Hyperparameter tuning is a very important word used in the deep learning field. Every deep learning model involves the use of different parameters which can be tuned to make it suitable to solve a particular problem. This can also be used to generate different models that will be part of the ensemble. So, by varying the different configurations of the different hyperparameters present in the model, a series of models that perform well on the training dataset can be generated. Examples of these hyperparameters are number of layers in the network, number of neurons in each layer, activation function, learning rate, etc. This will generate a group of ensemble models which will have the capability of having minimal overlap of predictions among themselves. There are a few hyperparameters present in the image captioning model. They include, the feature extraction model, the number of layers in caption generation model, the activation function, learning rate, normalization factor, dropout layers, and dropout rate.

D. Boosting Ensemble:

Boosting is a technique more complicated than stacking. In this approach, every data point in the training dataset is assigned a weight value and a subset of it is sampled based on the weights to be used to train a model. Initially all data points are given equal weights. The error in prediction on the training dataset is calculated after training every model. The weights of these wrongly predicted data points are increased based on the error rate, so that they have a higher chance of being sampled in the next subset. This process is repeated until all the ensemble members are trained. All these models are used to predict on a new image and these predictions are combined using one of the output combination techniques mentioned in the next section. This way data points that have not been learned correctly are learned by the next model. Hence a combination of predictions from all models can be guaranteed to give better predictions than a single model as every model learns from different data points and patterns.

E. Output Prediction Combination Variations:

When an ensemble of models is available, the process of prediction becomes complicated. The predictions of all ensemble model members have to be considered at every step so they can be combined in a form that can give better results. The ensembles on basic machine learning algorithms are combined using different techniques based on the problem being solved. In this work, we explore three different approaches to combine predictions from each model.

IV. Analysis

This section gives a detailed analysis of the different ensemble learning techniques used on image captioning and the performance variations that each of these techniques have shown. It shows how an ensemble of models is better in making predictions when compared to a single model trained on the entire dataset. Not just their benefits over a single model, but a comparison among these models to define the best ensemble technique that can be used to make predictions for image caption generation. During analysis, feature extraction for all types of ensemble learning has been performed using the Resnet50 pre-trained model as it has shown to provide the best prediction results during experimentation. Every ensemble of models and their results are shown in a single table representing the parameter combinations used in the model and the prediction accuracies. Model prediction results are shown for all of the ensemble models that have been tested out in this work and finally a conclusion is made based on these analytics so as to say which combination of ensemble model techniques are best suitable for image captioning.

A. Analysis of k-fold Cross Validation Ensemble

Different combinations of k-fold cross validation datasets have been chosen to experiment with different model architectures to understand the effects of ensemble learning on these datasets. The tables show the parameters of these combinations of models used and their individual bleu score accuracies along with the bleu scores of the whole ensemble.

K-fold Ensemble Member Combination-1: For this ensemble, the dataset is divided into 4-folds which requires 4 ensemble models to be trained on the 4 samples of datasets. Different combinations of two hyperparameters have been used in building the ensemble (shown in Table 1). Other hyperparameters namely the number of layers, number of neurons, learning rate and batches remain constant with values 6, 256, 0.001 and 256 respectively. It can be observed from Figure 2 that using ensemble-1 showed a significant increase in prediction accuracies over any of the individual models with all three combination methods. This shows that using combinations of LSTM and GRU model with different activations gives good results on image captioning.

K-fold Ensemble Member Combination-2: For this ensemble, dataset is divided into 4-folds (i.e., 4 ensemble models to be trained on 4 sample datasets). Different combinations of two hyperparameters have been used in building the ensemble (shown in Table 1). Other hyperparameters namely number of neurons, activation function, learning rate, and batches remain constant with values 256, Adam, 0.001 and 256 respectively. It can be observed from Figure 2 that using ensemble-2 showed a significant increase in prediction accuracies over any of the individual models with all three combination methods. This shows that using a combination of RNN models gives better results as they can capture different trends in data.

K-fold Ensemble Member Combination-3: For this ensemble, the dataset is divided into 3-folds (i.e., 3 ensemble models trained on 3 sample datasets). Different combinations of two hyperparameters have been used in building the ensemble (shown in Table 1). Other hyperparameters namely the RNN type, number of neurons, learning rate and batches remain constant with values LSTM, 256, 0.001 and 256 respectively. It can be observed from Figure 2 that using ensemble-3 showed very less increase in prediction accuracy with MV and AP methods over the individual models. But, using MMP in this case provided less accurate results than Model-2. The reason for this could be the use of Adagrad activation function. This states that Adagrad might not be a good function to use for image captioning. Another reason could be the number of ensembles used. Using only 3 ensemble members leads to capturing less trend variations in data.

Performing ensemble learning on k-folds of datasets has shown considerable amount of increase in accuracies when compared to the individual models. It can be understood from the charts provided that the first two ensembles performed better than any of their individual members by an amount of 1.6% in bleu-1 scores on an average. But, ensemble-3 did not show significant increase in accuracy. The reason for this is the number of k-folds chosen to perform ensemble learning. While the first 2 ensembles have datasets divided into 4-folds, the last ensemble has dataset split into 3-folds. This shows that small number of ensemble members will not be able to capture the different trends in the dataset. Hence, chosen a mediocre number of models to participate as ensemble members provides better results as shown in charts provided.

B. Analysis of Bootstrap Aggregation Ensemble

Bootstrap aggregation technique has been used to create subset samples from the dataset with replacements and used to train different models. The experimental results of applying bootstrap aggregation on image captioning are shown in Table 2. The bleu score of individual models and those of the combinations of these models are also presented.

Bootstrap Aggregation Ensemble Member Combination-1: For this ensemble, the dataset is sampled 8 times with each sample having a size of 3000 data points. Different combinations of three hyperparameters have been used in building the ensemble (shown in Table 2). Other hyperparameters namely the number of neurons, learning rate and batches remain constant with values 256, 0.001 and 256 respectively. It can be observed from Figure 3 that using ensemble-1 showed a good increase in prediction accuracies over all of the individual models using all combination methods. From this first ensemble, we can see that bootstrap aggregation works well with image captioning.

Bootstrap Aggregation Ensemble Member Combination-2: For this ensemble, the dataset is sampled 8 times with each sample having a size of 3000 data points. Different combinations of three hyperparameters have been used in building the ensemble (shown in Table 2). Other hyperparameters namely the number of neurons, learning rate and batches remain constant with values 256, 0.001 and 256 respectively. Figure 3 shows that using ensemble-2 showed an increase in prediction accuracies over all of the individual models using all combination methods. Based on the hyperparameters used, it can be inferred that using greater number of layers is affecting the combination prediction.

Bootstrap Aggregation Ensemble Member Combination-3: For this ensemble, the dataset is sampled 6 times with each sample having 4000 data points. Different combinations of three hyperparameters have been used in building the ensemble (shown in Table 2). Other hyperparameters namely the number of layers, number of neurons, learning rate and batches remain constant with values 6, 256, 0.001 and 256 respectively. Figure 3 shows that using ensemble-3 showed good increase in prediction accuracies over all individual models using all combination methods. It can be inferred from the result that using combinations of different RNN models with adam and rmsprop activations improves performance of image captioning. It can be concluded that using bootstrap aggregation ensemble technique will show significant increase in prediction accuracies with using the right combination of models. But, using a greater number of layers would affect the performance of the overall ensemble. Hence, a right combination of RNN models with activations having a smaller number of layers would give an increase in accuracy - bleu-1 score. The three models tested in this section showed an average increase of 1.7% in bleu-1 score.

C. *Analysis of Boosting*

In boosting, weights are assigned to each data point and the weights keep changing as more models are trained on them and error in predictions are calculated. A series of ensemble combinations have been tested out with the bleu scores of all of the ensembles presented in this section.

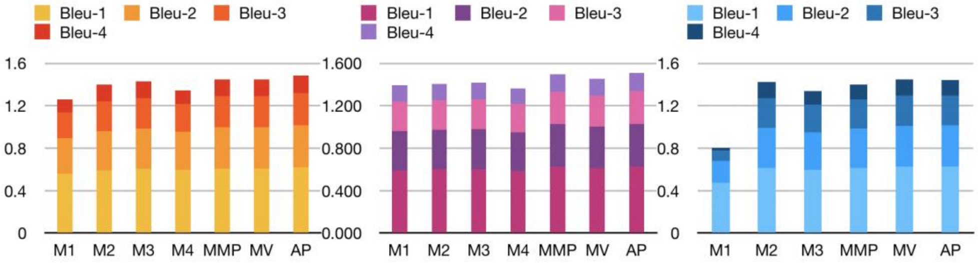
Boosting Ensemble Member Combination-1: For this ensemble, the dataset is sampled 4 times with each sample having 3000 data points. Different combinations of three hyperparameters have been used in building the ensemble (Table 3). Other hyperparameters namely the RNN type, number of neurons and batches remain constant with values LSTM, 256, and 256 respectively. Figure 4 shows that using ensemble-1 presents an increase in prediction accuracies over all individual models using all combination methods. A lot cannot be inferred from this ensemble as it has few ensemble members with random variations in hyperparameters.

Boosting Ensemble Member Combination-2: For this ensemble, the dataset is sampled 6 times with each sample having 4000 data points. Different combinations of three hyperparameters have been used in building the ensemble (shown in Table 3). Other hyperparameters namely the number of neurons and batches remain constant with values 256 and 256 respectively. Figure 4 shows that using ensemble-2 presents an increase in prediction accuracies over all individual models using all combination methods. This shows that using different combinations of activations and learning rates with the same RNN type could result in the models capturing different trends in data.

Boosting Ensemble Member Combination-3: For this ensemble, the dataset is sampled 7 times with each sample having 4000 data points. Different combinations of four hyperparameters have been used in building the ensemble (shown in Table 3). Other hyperparameters namely the number of layers, number of neurons, learning rate and batches remain constant with values 6, 256, 0.001 and 256 respectively. Figure 4 shows that using ensemble-3 showed an increase in prediction accuracies over all of the individual models using all combination methods. This shows that using different combinations of activations with different RNN types would also result in higher prediction accuracies. Boosting ensemble technique has shown best results among the three techniques that have been analyzed. All combinations methods have shown better performance than the individual models by 2.1% increase in bleu-1 scores on an average. One thing can be inferred from the models used for boosting. Using a constant learning rate of 0.001 for all ensemble members has better results with 100% guarantee.

Table 1. Ensembles with Corresponding Model Types and Bleu Scores for k-fold Cross Validation

	Model	**RNN type**	**Activation function**	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**
Ensemble-1	M-1	LSTM	Rmsprop	0.558	0.336	0.243	0.123
	M-2	GRU	Adam	0.593	0.370	0.280	0.156
	M-3	LSTM	Adam	0.607	0.382	0.286	0.156
	M-4	GRU	Rmsprop	0.597	0.360	0.258	0.133
	MMP	N/A	N/A	0.610	0.386	0.292	0.162
	MV	N/A	N/A	0.611	0.388	0.292	0.160
	AP	N/A	N/A	0.621	0.397	0.300	0.167
Ensemble-2	**Model**	**RNN type**	**No. of layers**	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**
	M-1	LSTM	6	0.590	0.374	0.280	0.153
	M-2	GRU	6	0.600	0.375	0.280	0.150
	M-3	SimpleRNN	6	0.603	0.378	0.282	0.157
	M-4	LSTM	8	0.584	0.363	0.270	0.145
	MMP	N/A	N/A	0.628	0.403	0.302	0.166
	MV	N/A	N/A	0.614	0.391	0.290	0.158
	AP	N/A	N/A	0.625	0.406	0.306	0.170
Ensemble-3	**Model**	**No. of layers**	**Activation function**	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**
	M-1	8	Adagrad	0.472	0.211	0.098	0.020
	M-2	6	Adam	0.613	0.382	0.280	0.150
	M-3	6	Adagrad	0.595	0.357	0.257	0.129
	MMP	N/A	N/A	0.614	0.375	0.271	0.143
	MV	N/A	N/A	0.624	0.388	0.284	0.154
	AP	N/A	N/A	0.626	0.388	0.280	0.147

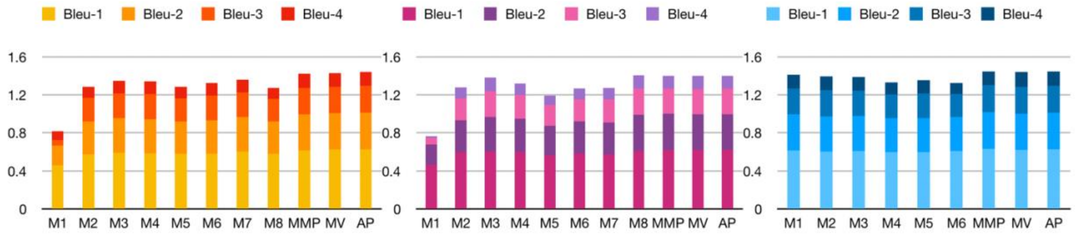


Ensemble-1 Ensemble-2 Ensemble-3

Figure 2. Bleu Score Comparision for k-fold Cross Validation Ensembles.

Table 2. Ensembles with Corresponding Model Types and Bleu Scores for Bootstrap Aggregation

	Model	RNN type	No. of layers	Activation function	Bleu-1	Bleu-2	Bleu-3	Bleu-4
Ensemble-1	M-1	LSTM	6	Adagrad	0.462	0.203	0.060	0.091
	M-2	LSTM	6	Adam	0.576	0.346	0.244	0.116
	M-3	GRU	6	Adam	0.594	0.360	0.260	0.131
	M-4	SimpleRNN	6	Adam	0.589	0.357	0.261	0.137
	M-5	LSTM	8	Adam	0.583	0.338	0.240	0.122
	M-6	GRU	8	Adam	0.583	0.350	0.256	0.134
	M-7	GRU	6	Rmsprop	0.603	0.362	0.259	0.134
	M-8	LSTM	6	Rmsprop	0.580	0.342	0.236	0.113
	MMP	N/A	N/A	N/A	0.616	0.380	0.277	0.147
	MV	N/A	N/A	N/A	0.626	0.383	0.276	0.143
	AP	N/A	N/A	N/A	0.626	0.387	0.280	0.147
Ensemble-2	**Model**	**RNN type**	**No. of layers**	**Activation function**	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**
	M-1	LSTM	8	Adam	0.465	0.212	0.073	0.016
	M-2	LSTM	6	Adam	0.595	0.337	0.233	0.111
	M-3	GRU	6	Adam	0.601	0.368	0.270	0.143
	M-4	SimpleRNN	6	Adam	0.600	0.352	0.247	0.121
	M-5	LSTM	8	Rmsprop	0.565	0.312	0.215	0.101
	M-6	GRU	8	Rmsprop	0.587	0.334	0.232	0.111
	M-7	GRU	6	Rmsprop	0.574	0.338	0.240	0.119
	M-8	LSTM	6	Rmsprop	0.611	0.380	0.273	0.140
	MMP	N/A	N/A	N/A	0.623	0.376	0.266	0.134
	MV	N/A	N/A	N/A	0.623	0.373	0.267	0.139
	AP	N/A	N/A	N/A	0.623	0.375	0.266	0.134
Ensemble-3	**Model**	**RNN type**		**Activation function**	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**
	M-1	LSTM		Adam	0.616	0.377	0.273	0.144
	M-2	GRU		Adam	0.602	0.371	0.274	0.147
	M-3	SimpleRNN		Adam	0.607	0.371	0.268	0.139
	M-4	LSTM		Rmsprop	0.599	0.354	0.253	0.127
	M-5	GRU		Rmsprop	0.596	0.361	0.260	0.135
	M-6	SimpleRNN		Rmsprop	0.609	0.356	0.244	0.117
	MMP	N/A		N/A	0.631	0.390	0.280	0.146
	MV	N/A		N/A	0.620	0.384	0.282	0.151
	AP	N/A		N/A	0.626	0.388	0.282	0.148



Ensemble-1 Ensemble-2 Ensemble-3

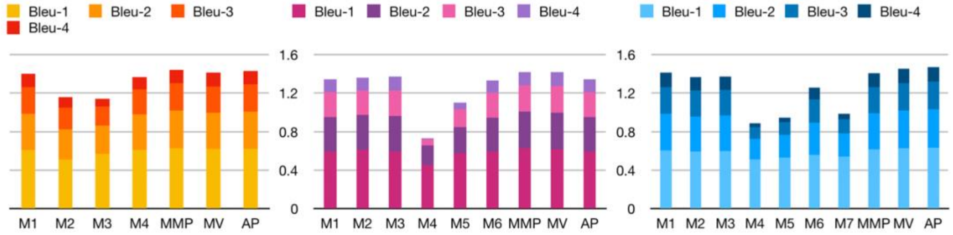
Figure 3. Bleu Score Comparision for Bootstrap Aggregation Ensembles

Table 3. Ensembles with Corresponding Model Types and Bleu Scores for Boosting

	Model	RNN type	No. of layers	Activation function	Learning rate	Bleu-1	Bleu-2	Bleu-3	Bleu-4
Ensemble-1	M-1		6	Adam	0.001	0.612	0.373	0.275	0.142
	M-2		6	Adam	0.0001	0.510	0.314	0.224	0.108
	M-3		10	Rmsprop	0.001	0.568	0.298	0.194	0.084
	M-4		6	Rmsprop	0.001	0.610	0.368	0.261	0.128
	MMP		N/A	N/A	N/A	0.630	0.390	0.280	0.143
	MV		N/A	N/A	N/A	0.621	0.378	0.272	0.140
	AP		N/A	N/A	N/A	0.625	0.386	0.278	0.143
Ensemble-2	**Model**	**RNN type**	**No. of layers**	**Activation function**	**Learning rate**	**Bleu-1**	**Bleu-2**	**Bleu-3**	**Bleu-4**
	M-1	Simple RNN	6	Adam	0.001	0.593	0.358	0.260	0.135
	M-2	LSTM	6	Adam	0.001	0.611	0.363	0.255	0.130
	M-3	GRU	6	Adam	0.001	0.595	0.365	0.270	0.143
	M-4	LSTM	6	Adam	0.0001	0.455	0.201	0.064	0.012
	M-5	LSTM	10	Rmsprop	0.001	0.576	0.270	0.184	0.072
	M-6	LSTM	6	Rmsprop	0.001	0.594	0.354	0.255	0.130
	MMP	N/A	N/A	N/A	N/A	0.628	0.380	0.271	0.140
	MV	N/A	N/A	N/A	N/A	0.618	0.380	0.275	0.146

	AP	N/A	N/A	N/A	N/A	0.593	0.358	0.260	0.135

	Model	RNN type	Activation function	Bleu-1	Bleu-2	Bleu-3	Bleu-4
Ensemble-3	M-1	SimpleRNN	Adam	0.607	0.380	0.278	0.148
	M-2	LSTM	Adam	0.591	0.366	0.270	0.140
	M-3	GRU	Adam	0.599	0.367	0.266	0.140
	M-4	GRU	Adam	0.511	0.213	0.122	0.044
	M-5	LSTM	Adam	0.532	0.234	0.134	0.043
	M-6	LSTM	Rmsprop	0.561	0.335	0.240	0.120
	M-7	LSTM	Adam	0.542	0.244	0.144	0.057
	MMP	N/A	N/A	0.618	0.374	0.273	0.142
	MV	N/A	N/A	0.629	0.391	0.284	0.151
	AP	N/A	N/A	0.634	0.400	0.287	0.151



Ensemble-1 Ensemble-2 Ensemble-3

Figure 4. Bleu Score Comparision for Boosting Ensembles

D. Sample Captions Generated for Images in Train Dataset

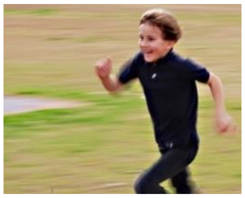


*Figure 5. Sample Image in Test Dataset-1 **(Example 1)***

Best Model – “two young boys are playing on the grass”

k-fold Ensemble – “boy in blue shirt is running in the grass”

Bootstrap Aggregation Ensemble – boy in red shirt is running on the grass”

Boosting Ensemble – “boy in blue shirt is running on the grass”

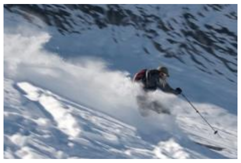


*Figure 6. Sample Image in Test Dataset-2 **(Example 2)***

Best Model – “skier is walking through the snow”

k-fold Ensemble – “skier is skiing down snowy hill”

Bootstrap Aggregation Ensemble – “skier is in the snow”

Boosting Ensemble – “man in red jacket is skiing down snowy hill”

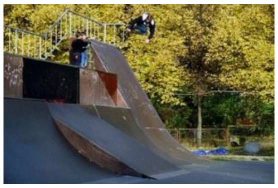


*Figure 7. Sample Image in Test Dataset-3 **(Example-3)***

Best Model – “man in red shirt is jumping down ramp”

k-fold Ensemble – “skateboarder is jumping off of the railing”

Bootstrap Aggregation Ensemble – “man is jumping down the wall”

Boosting Ensemble – “skateboarder is jumping off ramp”

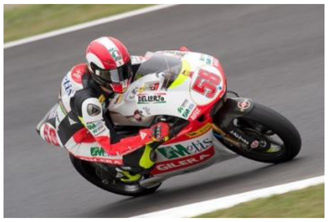


*Figure 8. Sample Image in Test Dataset-4 **(Example 4)***

Best Model – “man in yellow helmet is riding bike on the road”

k-fold Ensemble – “motorcycle racer is riding motorcycle”

Bootstrap Aggregation Ensemble – “man in red helmet is riding the bike”

Boosting Ensemble – “man in red and white helmet is riding bike on the track”

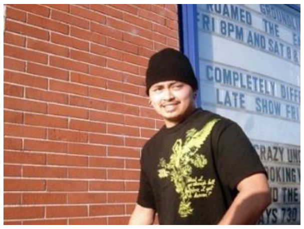


*Figure 9. Sample Image in Test Dataset-5 **(Example 5)***

Best Model – “man with sunglasses and sunglasses”

k-fold Ensemble – “man in black shirt and black hat is standing in front of an old building”

Bootstrap Aggregation Ensemble – “man in black shirt and black shirt is standing on the street”

Boosting Ensemble – “man in black shirt is standing on the street”

E. Combined Analysis of all methods

It can be observed from Table 4 that all three ensemble techniques have clearly shown improvements in prediction accuracies over the best individual model. Among these ensembles, boosting has shown the highest increase. The captions generated for images in the training dataset by the ensemble models also clearly show the generation of better captions.

Table 4. Comparison of all Ensemble Methods with the Best Individual Model

Model	Bleu-1	Bleu-2	Bleu-3	Bleu-4
Best Model	0.616	0.377	0.273	0.144
k-fold	0.628	0.403	0.302	0.166
Bootstrap Aggregation	0.631	0.390	0.280	0.146
Boosting	0.634	0.400	0.287	0.151

V. Conclusion and Future Scope

The idea of using ensemble learning techniques on image captioning problem has been proven to be advantageous over using a single model. With the results presented in the analysis section, it can be clearly seen that forming an ensemble with the right combination of models would show significant improvements in performance. All the three ensemble learning techniques used in our work promised better predictions in almost every case. On an average all the techniques showed an increase in overall accuracy of 2.2% over the best individual model in the ensemble. An increase in accuracies by 2.2% is considered very crucial for deep learning models. Though training many models is computationally expensive when compared to training just one model, the use of a sampled dataset for training each model reduces this expense by a huge amount. Hence, the compute cost would be very close to that of training a single model on the entire dataset. Also, using sampled datasets is a crucial part of ensemble learning which would result in faster training of the models. Choosing the right number of ensemble members to balance the computational cost with increase in accuracies is critical in this case. From the analysis done in this work, using 4-5 ensemble members and training each of the members with 25% of the entire dataset would be sufficient to attain significant increase in prediction accuracies. Ensemble learning thus helped in solving the problem of high variance occurring in deep neural networks. Different models used as ensemble members learned different patterns from the training dataset. Though each of them individually could not generalize well on the data, when combined together had the advantage of knowing diverse patterns in data learned by each of them. Another advantage here is using a sampled dataset. All models could observe only subsets of the training data and so each of them would generate varied predictions. These diverse predictions when combined in a standard way generated the desired results, thus decreasing the problem of high variance.

As part of future work, different variations to the traditional model for solving image captioning can be used as model members for the ensemble. This way a more differing set of patterns can be learned by different models which when combined would result in giving even better predictions. So, the more variations in patterns that the ensemble members can learn the more would be the efficiency of combined predictions. Another improvement could be using a larger dataset and training the ensemble models with sampled datasets of the training set. This would help the different models to learn and capture different patterns from subsets and together make stronger predictions.

VI. References

[1] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei, “Incorporating Copying Mechanism in Image Captioning for Learning Novel Objects,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6580-6588, 2017.

[2] Xuedan Du, Yinghao Cai, Shuo Wang, and Leijie Zhang, “Overview of Deep Learning”. *IEEE 31st Youth Academic Annual Conference of Chinese Association of Automation*, pp. 159-164, 2016.

[3] Senmao Ye, Junwei Han, and Nian Liu, “Attentive Linear Transformation for Image Captioning,” *IEEE Transactions on Image Processing*, 27(11), pp.5514-5524, 2018.

[4] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi, “Understanding of a Convolutional Neural Network,” *IEEE Intl. Conference on Engineering and Technology (ICET)*, pp. 1-6, 2017.

[5] [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[6] Aya Abdelsalam Ismail, Timothy Wood, and Hector Corrada Bravo, “Improving Long-Horizon Forecasts with Expectation-Based LSTM,” arXiv preprint arXiv:1804.06776, Apr, 2018.

[7] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv:1409.1556 (2014).

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, “Rethinking the Inception Architecture for Computer Vision”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826, 2016

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, *IEEE Conf. on Computer Vision & Pattern Recognition,* pp. 779-788, 2016.

[10] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei, “ Boosting Image Captioning with Attributes”, *IEEE International Conference on Computer Vision*, pp. 4894-4902, 2017.

[11] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Chourville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, *International Conference on Machine Learning*, pp. 2048-2057, 2015.

[12] [Online]. Available: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

[13] Junyi Xu, Li Yao, Le Li, “Argumentation Based Joint Learning: A Novel Ensemble Learning Approach,” PloS one, 10(5), 2015.

[14] [Online]. Available: https://machinelearningmastery.com/en semble-methods-for deep-learning-neural-networks/

[15] [Online]. Available: https://www.datascience.com/resources/notebooks/word-embed dings-in-python

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, “Bleu: A Method for Automatic Evaluation of Machine Translation,” Proceedings for the *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311-318, 2002.

[17] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. “Show and Tell: A Neural Image Caption Generator,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156-3164, 2015.

[18] Kenneth Tran, Xiaodong He, Lei Zhang, Jian Sun, Cornelia Carapcea, Chris Thrasher, Chris Buehler, Chris Sienkiewicz. “Rich Image Captioning in the Wild,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 49-56, 2016.

[19] Urminder Singh, Sucheta Chauhan, A. Krishnamachari, Lovekesh Vig. “Ensemble of Deep Long Short Term Memory Networks for Labelling Origin of Replication Sequences,” *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1-7, 2015.

[20] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, Sanghoon Lee. “Ensemble Deep Learning for Skeleton-based Action Recognition using Temporal Sliding LSTM Networks,” *IEEE International Conference on Computer Vision (ICCV)*, pp. 1012-1020, 2017.