Enhancing Image Caption Quality using an Ensemble of Deep Learning Models

MSc Research Project
MSc Data Analytics

Harisankar
Student ID: X22247564

School of Computing
National College of Ireland

Supervisor: Jorge Basilio

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Harisankar		
Student ID:	X22247564		
Programme:	Msc. Data Analytics	**Year:**	2024-2025
Module:	Research Project		
Supervisor:	Jorge Basilio		
Submission Due Date:	29-01-2025		
Project Title:	Enhancing image caption quality using an ensemble of Deep Learning models		
Word Count:	8219	**Page Count**:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Harisankar

Date: 29-01-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Image Caption Quality using an Ensemble of Deep Learning Models

Harisankar
X22247564

Abstract

This research analyses of how effective it is to combine various CNN architectures with LSTM models and ensemble methods with the aim of enhancing image captioning performance. The proposed study analyses three CNN architectures: VGG16, ResNet50 and Xception, and their usage in CNN-LSTM architectures for captioning tasks. Also, ensemble methods like Bagging and Boosting were employed to take advantage of those base models. The evaluation metrics used were BLEU and METEOR scores. The results demonstrated that ResNet50 was the best performing model overall, since it attained the highest BLEU and METEOR scores, showing that its generated captions are most contextually relevant. Quite significant results were achieved from the model VGG16 and Xception, but they still had weaknesses in terms of dealing with more complicated multi-word phrase. Bagging seems to have somehow benefited from the diversity of the models, but it created grammatical mistakes in models that could be due to combination of dissimilar outcomes. Boosting, in contrast, was able to produce good results by assigning weights and iterating over base models, which in turn generated captions that are relevant to the images.

Keywords – Convolutional Neural Network, image caption, ensemble methods, Long Short-Term Memory

1 Introduction

Image captioning, which is the process of giving pictures text explanations, is hard and needs computer vision and natural language processing to work together smoothly. It has gotten a lot of attention lately because it can be used for so many different things. For example, it can help people who are blind or have low vision, it can automatically annotate material for multimedia platforms, and it can make image search engines better. Image captioning improves accessibility across many areas by turning visual content into descriptive text. This makes it easier for people to interact with computers.

Template-based models or retrieval-based methods were used in the early days of image captioning. These methods were not flexible or accurate enough to make dynamic and context-sensitive captions. Deep learning changed the field in a big way. Convolutional neural networks (CNNs) are great at extracting details from images, and recurrent neural networks (RNNs), especially Long Short-Term Memory (LSTM) networks, are great at making words that make sense (Bhatt et al., 2023). Even though each of these models has come a long way, their results don't always show the full range of details of an image's context. Because of how they are built, cutting-edge CNN architectures like ResNet50, VGG16, and Xception have shown great success in tasks that require extracting features. For instance, ResNet50 uses residual learning to solve the vanishing gradient problem, VGG16 focusses on simplicity with uniform design, and Xception uses depth wise separable convolutions to make computations faster (Santi et al., 2024). Even with these improvements, using only one CNN-LSTM pipeline can lead to poor caption quality when used with different datasets or complicated pictures.

These issues might be able to be solved with ensemble learning. Bagging and boosting are techniques that use the best parts of multiple models to make the results of a single model better. Ensemble methods are used a lot in other areas of machine learning, but no one has looked into how they can be used to caption images yet. The traditional image captioning relies on a single deep learning architecture. CNNs detect features in images, while word generation is done by LSTM networks (Jain et al., 2019). Though ResNet50, VGG16, and Xception models work well separately, they show poor performance in large datasets or diverse scenarios (Islam et al., 2020). This research has recognized the lacuna in this domain due to the integration of ensemble methods in overcoming such challenges in the task of image captioning.

The main goal of this project is to determine which ensemble technique performs best in improving the quality of image captions generated by CNN-LSTM-based models. To achieve this aim, the project has the following objectives:

- To build and test unique CNN-LSTM models using the ResNet50, VGG16, and Xception architectures.
- To incorporate ensemble methods like bagging and boosting to combine the outputs of these models.
- To evaluate these models' using well-known rating systems, such as BLEU and METEOR.
- To determine the ensemble method that improves the meaning and context of captions the most.

Research Question: How an ensemble of different deep learning models like Convolutional Neural Network, Long Short-Term Memory and its different architectures can be effectively combined to enhance the quality and accuracy of generated image captions?

Increasing the quality of image caption creation has big effects on both business and academia. Strong and accurate titles have a huge impact on everything from making videos easier for visually impaired people to use, to improving multimedia management systems. The latest CNN architectures and ensemble learning methods are brought together in this study, which adds new knowledge to the field. Also, the results will give researchers evidence-based advice on how to make image captioning systems work better.

This research looks at three CNN architectures—ResNet50, VGG16, and Xception that are used as the base for CNN-LSTM models to extract features. Then, ensemble methods like bagging and boosting will be used to combine the results. Commonly used measures, such as BLEU and METEOR will be used to judge how well these models work. As a result of the analysis, the research gap will be filled by finding the ensemble method that improves caption quality the most.

The structure of the report is presented as follows. Section 2 discusses the theoretical foundations of CNNs, LSTMs, and ensemble learning, and critically reviews prior studies in image caption generation. Section 3 describes the methodology being followed for building this project. The architecture and design of the model is specified in section 4. In section 5, details of experimental setup, including dataset selection, model development, and evaluation procedures are covered. Section 6 presents the findings and analysis of the individual models and ensemble techniques. Section 7 summarizes key insights, outlines limitations, and provides suggestions for future research.

2 Related Work

Image captioning has changed a lot over the years, and deep learning models are now the most important tool for making good captions. This part talks about the ideas behind CNNs, LSTMs, and ensemble learning, as well as the latest developments in these fields. It focusses on their pros and cons. The goal is to give a full review of the literature, look at any gaps critically, and show why this work is important.

2.1 Convolutional Neural Network (CNN) in Image Captioning

Convolutional Neural Networks (CNNs) are very important to image captioning systems because they help a lot with getting spatial and hierarchical information from pictures (Fudholi et al., 2022). These networks have revolutionized the field, enabling robots to analyze and interpret visual information with a level of accuracy that had never been witnessed before. ResNet50, VGG16, and Xception are architectures distinguished by very special design principles, known for generalizing well in a very large number of scenarios, like picture captioning.

Sharma and Singh (2021) created ResNet50, which uses residual learning to improve deep network vanishing gradients. Skipping links enhances gradient flow in ResNet50. Complex patterns can be taught without slowing down. This architecture extracts fine-grained features better, making it accurate for jobs like picture captioning. Building strong ResNet50 is difficult, memory and processor-intensive. This is not suitable for particularly when working with huge or in real-time systems, as Cai et al. (2022) noted.

Popular design VGG16 was created by Sharma et al. (2022). This model uses simple uniform convolutional layers with modest fields for feature extraction. Its consistent layout improved generalization across datasets, which also made it a good choice for many image processing tasks (Gkelios et al., 2021). Although it has many benefits, VGG16 is difficult to be used in practice due to its memory and processing requirements. The VGG16's performance is sometimes less efficient than current designs, according to Carlos (2024). These results suggest fixing or adding group learning.

Chollet (2017) developed Xception model, which speeds processing with depth wise separable convolutions. The Inception model is optimized for performance and reduced parameters with this architecture. Xception is good for photo captioning feature extraction since it can swiftly process big, complex datasets. Jinsakul et al. (2019) prove that hyperparameter tuning impacts Xception, therefore, by a large margin, making it less user-friendly, as high quality data are compulsory for it, which lowers its reliability in cases when noisy or partial datasets are given.

While each CNN design has its strengths, good performance does depend upon the information and the tasks' difficulty. ResNet50 does better in extracting complex features from high-resolution pictures, while Xception does better when the processing power is not strong (Zhang et al., 2024). VGG16 has good balance but needs more resources. Ensemble designs can overcome CNN limitations through assembling the best aspects of a large number of designs for accurate and reliable image captioning.

Adding the attention mechanisms helps to focus the CNNs on the relevant parts of images for caption generation. Wu et al. (2023) showed that CNN-based models with attention mechanism can understand the context better. Found that CNNs work well but need to integrate with other approaches for improving accuracy and applicability across datasets.

Image captioning systems require CNNs architectures like ResNet50, VGG16, and Xception to extract the features embedded in the images. Each design has merits, but their limits require ensemble techniques to improve the overall. Ensemble learning fills in gaps with these designs' strong feature extraction ability, making captions more accurate and contextually rich.

2.2 Ensemble Learning

An effective machine learning technique called "ensemble learning" takes the best parts of many models and combines them to make performance better. Although it is often used for classification and regression, Alzubi et al. (2020) say that its promise for describing pictures is not fully explored. For picture labelling, ensemble methods like bagging and boosting are useful because every model sees and understands visual data in its own way. One can plan how to combine the traits of different models with these methods, which makes captions more accurate and meaningful.

Fan (2009) suggests "bagging" guesses from models that were trained separately to cut down on variation and boost performance. Bagging helps lower picture captioning overfitting by training models on many groups of data. Dong et al. (2017) tried bagging CNN-based models for picture captioning and found that BLEU scores improved dramatically. Because it trains multiple models at once, bagging requires a lot of processing resources, the study found. For places with lots of data or low resources, this can be problematic.

A common ensemble strategy, boosting fixes errors caused by previous models sequentially. Ganaie et al. (2022) found that boosting strengthens weak learners. By highlighting hard to guess examples, boosting can improve picture captions. Their work showed that boosting on short datasets might overfit, highlighting the need to properly alter parameters. Ensemble learning reduces the variability of output model, making it useful for picture captioning (Zhang et al., 2024). Ensemble approaches benefit from CNN designs like ResNet50, VGG16, and Xception, which extract features better. ResNet50 gathers little details, Xception optimizes computing, and VGG16 generalizes well. Ensemble learning creates semantically deep and contextually appropriate captions by mixing structures (Ardabili et al., 2020).

Despite their merits, ensemble approaches are difficult to apply in photo captioning. Many research exclusively examines single CNN-LSTM models, ignoring ensemble approaches' robustness and flexibility. Ensemble methods are difficult to compute and integrate, which causes issues. Ensemble models work better than individual models, but they require a lot of resources, thus they may not be suitable for real-time applications (Zhou et al, (2021). Group learning requires advanced assessment methods like BLEU and METEOR (Berger et al., 2024). Kotu and Deshpande (2019) found that ensemble techniques continuously improve scores across multiple parameters, resulting in captions that closely relates to human references. These enhancements generally increase training and processing time, highlighting the importance of optimized ensemble systems.

Ensemble learning improves picture captioning by combining the best features of numerous models. Bagging and boosting increase subtitle semantics and context (Stefanini et al., 2022). To improve performance, computational difficulties, resource demands, and overfitting must be addressed. We can avoid single model issues by using ensemble learning and best CNN structures.

2.3 Evaluation metrics

The image captioning models need to be evaluated using the quantitative methods by comparing the generated captions with human references regarding correctness and relevance (Sharma et al., 2023). BLEU and METEOR scores are being used in image captioning for capturing linguistic and semantic features. These measurements provide a lot of useful information, but they also have issues that make it even more important to use additional evaluation methods, including human evaluations, to properly evaluate model performance.

Khandelwal (2020) studied the BLEU (Bilingual Evaluation Understudy), used it to measure n-gram overlap between a generated caption and one or more reference captions. The score depends on how similar the generated text is to the reference text. More points mean greater results. Because it is simple and accurate for word-to-word checks, BLEU has become a standard (Datta et al., 2022). Still, it has been criticized for not considering semantic meaning and not handling synonyms. BLEU may not consider titles that employ synonyms or paraphrases, even when they convey the same meaning, which could lead to incorrect judgements.

When Chauhan and Daniel (2022) made METEOR (Metric for Evaluation of Translation with Explicit Ordering), they fixed some problems with BLEU by adding stemming and word matching. METEOR rates the accuracy and memory of language while also taking order of the word into account, which results in a more fair rating of language quality. Because of this, it works especially well for picture captioning jobs that value ease of reading and flow. But METEOR uses a lot of resources and might put memory over accuracy, which could lead to varied results for long captions (Lee et al., 2023).

Methods like BLEU and METEOR quantify model performance, but they don't account for caption creation's subjective and creative features (Sai et al., 2023). Factually correct captions without originality or emotion may score well using these criteria. Because of this, automatic methods still need human inspection. Human reviews assess model performance by considering usefulness, consistency, and linguistic variety. Human review is time-consuming and may result in prejudices. In recent years, hybrid evaluation systems have combined computerized and human assessments. These solutions take advantage of automated measures to benefit while resolving their drawbacks. This makes for a more complete evaluation system.

BLEU and METEOR are some of the evaluation methods that can be used to rate the quality of picture captions. There are pros and cons to each of these. These measurements help us see how different things are, but they often miss subjective aspects like imagination and the fullness of the situation. This means that to make the process of reviewing picture captioning systems better, we need to mix automated tests with human reviews and look into new hybrid frameworks. The fair method makes sure that the reviews of the models are more accurate, which helps the creation of captioning systems that meet both technical needs and user standards.

2.4 Gaps in Literature

There are still some issues with the current study, even though image captioning has come a long way. With CNN-LSTM systems like those that use ResNet50, VGG16, or Xception, a lot of research is done on smaller groups. Sometimes these designs aren't resilient and have trouble applying them to new datasets, but they work well when they do. Even though ensemble learning has made big steps forward in other areas, not nearly enough study has been done on how to use techniques like bagging and boosting to combine model benefits (Bhatt et al., 2023) in picture captioning. Also, most of the research we do now uses a small group of review metrics, like BLEU and METEOR. These metrics are helpful, but they don't really show the subjective qualities of creativity and contextual depth that make captions great. The fact that automated measures are used too much and ensemble approaches aren't used enough shows how important it is to do research that creates and tests ensemble frameworks for image captioning using both standard evaluation metrics and human evaluation methods to fully fix these problems.

3 Research Methodology

The methodology for this research for enhancing image captioning through CNN-LSTM models and ensemble learning with different CNN architectures is structured into several key components. Each component is designed to systematically address the research objectives outlined in the introduction.

3.1 Dataset

This research uses Flickr8k dataset that contains 8,091 images and a total of 40,455 captions. The dataset was downloaded from Kaggle with was available for public and have CCO public domain license. There will be many captions for each image model to fully understand diverse descriptions and viewpoints, so enriching the quality of generated captions. The dataset encompasses a diverse range of subjects, including individuals, animals, and different scenarios, offering a solid basis for training models capable of generalizing across many settings. Efficient data management and preprocessing can be achieved by using this structured format thereby simplifying the creation of mappings between images and captions. The collection is also seen as a standard in the field for judging picture labelling methods, which makes it possible to make important comparisons with recent research. The fact that there are many reference captions works well with evaluation tools like BLEU and METEOR which are necessary for checking the quality of captions. The Flickr8k dataset is perfect for building and testing advanced systems that use ensemble learning to add captions to pictures because of the way it is structured.

3.2 Data Preprocessing

1. **Importing Libraries:** It is essential to have libraries like Keras, Numpy and NLTK. These libraries are very important for data management, model building and even executing natural Language processing related tasks.
2. **Loading Captions:** Captions are fetched from a given location on disk from a text file captions.txt. The first line containing headers is removed and rest of the documents are assigned into string variable.
3. **Mapping Images to Captions**: A caption is stored corresponding to an ID of the image, where the caption contains the description of the image. Each image has five corresponding captions. The algorithm proceeds to check each line from the document and sees each comma as a separator and retrieves the image ID and the captions. The extension in the image id representing the file type is removed. These captions are then stored into a list.
4. **Pre-processing Captions:** A function called clean_captions is built to perform the task of cleaning and normalizing of captions. These includes changing of all the text to lower cases, adding or removing unwanted characters to the text being used, changing unwanted spaces to one space and adding beginning (startseq) and ending (endseq) tokens for individual captions.
5. **Tokenization**: Keras Tokenizer is used to tokenize the cleaned captions. The process of changing words to the numbers that are their replacements is called tokenization, each words will be then represented by an integer number. Vocabulary is created using these words. Further the vocabulary size will be used for the purpose of training the model.

6. **Calculation of Maximum Caption Length**: The maximum length of captions is calculated to maintain the consistency of the inputs. This will ensure that inputs required for the model is passed correctly.
7. **Train-Test Split**: The divided into training and testing sets in the ratio 90 to 10. Splitting the data into training and testing will ensure the model will be trained for learning all the features from the images and the testing dataset can be used to evaluate the prediction of the model.
8. **Data Generator Development**: A data generator function is developed to produce batches of data throughout model training. This function converts photos and their associated captions into input-output pairs appropriate for training CNN-LSTM models. These methods jointly guarantee that the dataset is prepared for efficient model training and assessment in image captioning, hence improving the model's capacity to provide precise and contextually appropriate captions. Data generator helps to reduce the computational load of the system by considering them in different batches of the given size.

3.3 Modelling

3.3.1 Model Architecture selection

The architectures selected for this research, ResNet50, VGG16 and Xception are very effective for image processing tasks and also so in image captioning. It was observed that ResNet50 solves the challenge of vanishing gradients and this allows training of very deep networks while maintaining performance and its deep residual learning architecture is chosen for this purpose (Shafiq & Gu, 2022). VGG16 has easily been chosen because it performs well in picture processing due to its ability to take spatial hierarchies in images into consideration (Gayathri et al., 2023). Xception improves computing efficiency while maintaining high accuracy in feature extraction, is implemented using depth wise separable convolutions (Usman Ghous et al., 2024). The innovative aspect of this work is the application of modern CNN models with the aim of enhancing captioning models through better feature extraction and their further combination with ensemble methods.

3.3.2 Feature Extraction

In this study, three powerful CNN models, namely ResNet50, VGG16 and Xception, are used to extract features from images. Images are modified to meet the size requirements of the CNN models which is 224x224 for ResNet and VGG16, and 299x299 for Xception. Where images of the same category are analyzed and models are used to extract features from images, ResNet50’s residual learning framework enables the model to learn profound features, VGG16’s simple but effective architecture allows it to learn the spatial hierarchies, and finally, Xception, allows for more complex feature extraction using depth wise separable convolutions. These characteristics can adequately characterize the visual content and are essential to generating good quality captions (Alzubi et al., 2020). The characteristics are sequenced for later use in training the LSTM model for caption generation from these characteristics. This feature extraction method is beneficial to produce better quality of the captions generated from images in the image captioning task.

3.3.3 Model Building

The base architecture of the model is a CNN-LSTM model which combines the feature extraction function from CNN and sequential text generation using LSTM. This helps the project for extracting information from the image and generating meaningful captions. Three CNN networks ResNet50, VGG16 and Xception are used. The LSTM architecture is excellent for learning sequential input over a long time which is suitable for natural language understanding and generation. The first one is an embedding layer that transforms word indices into dense vectors to enhance the model's understanding of word meanings. After that, one or more LSTM layers that process word sequences are added to learn the relation of the caption and its context. To reduce the overfitting of training, dropout layers are incorporated into the design in order to enhance performance. At last, it moves to a dense layer that estimates the next word in relation to the LSTM layer features, that predicts the caption sentence. This output layer uses the softmax activation function that yields a vocabulary probability distribution so that the model picks the next word suitably. An in-house data generator creates batches of images with captions for maximum memory representation and model building. It helps the system to reduce the computational load and avoid crashes of system. A multi-class classification loss function such categorical cross entropy is optimized during the training and performance is evaluated using BLEU and METEOR. This model building strategy seeks the use of LSTMs' sequential leaning to build a model that would convert the visual features into relevant captions.

3.3.4 Ensemble Methods

Ensemble methods have been introduced in the research to improve the quality of captions generated by single models using ResNet50, VGG16, and Xception architectures. The two main ensemble techniques used are Bagging and Boosting, both having different benefits in enhancing the model performance.

Bagging or Bootstrap Aggregating is a technique of training multiple instances of the model on different subsets of the training data set. It reduces the variance by taking average of the predictions from individual models, which in turn produces more stable and accurate outputs. In picture captioning, Bagging can be used to reduce overfitting by ensuring that the model's prediction is not overly dependent on any one training instance. Research shows that Bagging can greatly enhance results on many machine learning tasks using the diversity of a combination of models. On the other side, Boosting is another powerful ensemble method. Here, the approach relies more on iterative model training.

Each subsequent model is trained to address the limitations of the previous model, thus improving overall accuracy. Boosting improves the performance of the model by focusing on instances that have been misclassified; in this way, the model learns from errors and can correct and refine its predictions. Especially beneficial in generating high-quality captions that precisely describes the information present in an image. Ensemble methods have been illustrated through studies to largely raise the predictive accuracy in several other fields as well.

3.4 Evaluation Metrics

The standard metrics used for evaluation of image captioning is BLEU scores. It compares the predicted captions with one or more reference captions. METEOR score is also being considered in the project for testing the results. It is calculated using the average of precision and recall. High score value shows the model is able to predict the captions that are close to human provided captions.

There are BLUE-1, BLEU-2, BLEU-3 and BLEU-4 scores which compares the overlapping of 1,2,3 or 4 words respectively. A score closer to 1 shows the model is accurately able to predict the captions and a score closer to 0 depicts the deficiency of the model. A BLEU score value greater than 0.4 proves that the generated captions from the model is meaningful to the context of image. Achieving a score greater than 0.4 proves the model is able performing well and can be used for image captioning functionality. Apart from quantitative evaluation, human evaluation also can be considered for image captioning models. Hence the quality of captions generated can be evaluated using BLEU score, METEOR and also by human judgements for testing individual images.

4 Design Specification

The main architectures and components that are utilized in this project is discussed in this section. The model developed for image caption generation is integrated using CNN and LSTM models. CNN can be used for feature extraction of images and LSTM for generating sequential texts representing the image. Different types of architectures like Resnet50, VGG16 and Xception were chosen for the CNN model. The architecture for image caption generation is built effectively to combine the image understanding using natural language processing, producing meaning captions for the provided input image.

4.1 Convolutional Neural Network (CNN)

CNN is the component that acts as image feature extractor. CNN have several convolutional layers which can analyse spatial representations. It can be used to process raw image data and produce high-level feature representation. It captures the important features of the images which are crucial for image caption generation. CNN also reduces the number of spatial features ensuring that the important features are conserved. Convolutional layers will be able to detect both simple and complex features in different layers. The output of convolutional layers will be a feature map which correspond to different aspects of the image. Rectified Linear Unit (ReLU) activations functions is applied after convolutional layer to maintain the non-linearity in the feature. ReLU clips the negative integers to 0 and keeps the positive values unchanged to maintain non-linearity in images (Kuo 2016). Pooling layers then reduce the number of spatial dimensions which helps to decrease the computational load of the model while preserving the important features. In the fully connected layers the output features are flattened and transformed to 1-Dimensional vector and passed through the dense layers. The last layer of fully connected network is then integrated with the LSTM model for decoding the extracted features to generate meaningful text.

Multiple CNN architectures like Resnet50, VGG16 and Xception have been explored in this project. Each of these architectures have its unique way of feature extraction and performance in image captioning tasks.

1. **Resnet50**: It has deep network with 50 layers which is equipped with skip connections to overcome the vanishing gradient problem. Resnet50 model, pre-trained on ImageNet dataset, can be used and the fully connected layer to be replaced with the feature extraction layer. The extracted features will be transferred through a dropout layer to prevent overfitting. The extracted features can be then passed through the dense layer to reduce the dimensionality which will align with the input requirements of LSTM model.
2. **VGG16**: It is a 16 layer network known for its simplicity and uniform architecture. The model uses small convolutional kernels and focus on increasing the depth of layers whilst keeping the size of features manageable. VGG16 is pre-trained model on ImageNet, which can be used to extract the feature of images. The output of VGG16 model is a vector of 4096 dimension. It will be processed through a dropout layer to prevent overfitting of data. The extracted features are then processed through a dense layer to reduce the dimension which can be used for the LSTM layer. VGG16 integrated with the LSTM model and merged with text embedding layer for processing.
3. **Xception**: Xception model is built upon inception model by introducing depth wise separable convolutions. This pre-trained model can be used for feature extraction and the processed data will be converted to a vector dimension of 2048. The processed data will be then passed through dropout layer and dense layer similar to other CNN models.

4.2 Long Short-Term Memory (LSTM)

LSTM is used for decoding the extracted features from CNN to generate sequential words relevant to the image. LSTM is a type of Recurrent Neural Network, which resolves the vanishing gradient problem. It can be used to generate image captions as it does not have the short-term memory issues while backpropagation in other RNN models. It is also good for handling temporal dependencies which makes LSTM suitable for this project. LSTM receives two inputs that are text embeddings and image feature vector. It processes the text data which can convert the words into a dense vector of 256 dimensional, later processed through a dropout layer to prevent overfitting of model. Masking mechanism is incorporated in the layer to remove the padding tokens available in text data. LSTM layer with 256 units dimensions is available to produce sequential output by also capturing the temporal dependency in the text data. It have memory cell structure with input gate, output gate and forget gate. The input gate will decide which data from the input and last input needs to be considered for processing. The forget gate decides which data needs to be removed from the memory and output gate chooses the data that is needed for the next step. The final dense layer, which has a softmax activation function will be predicting the output captions from the available vocabulary which is relevant to the input image.

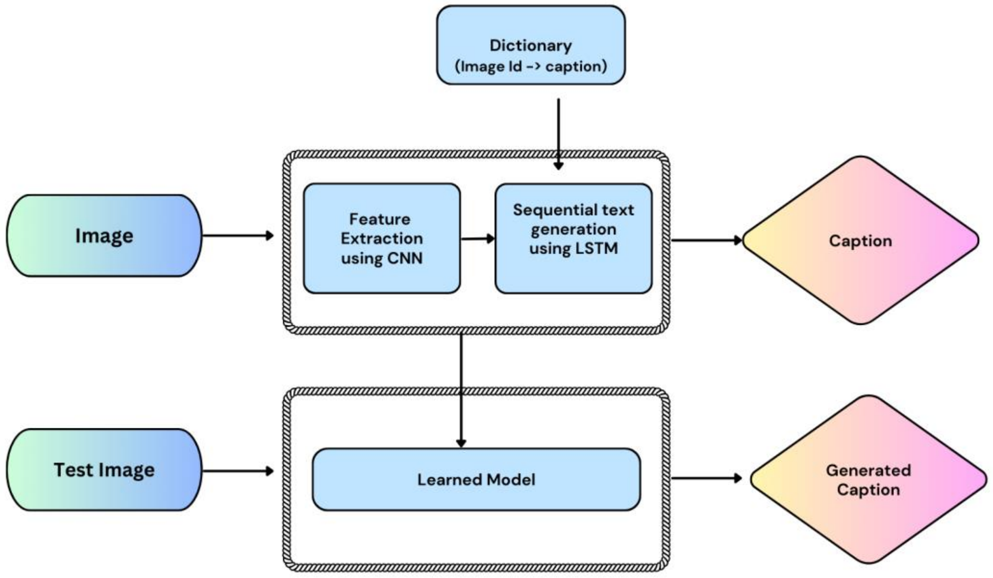


Figure 1: Block diagram of CNN-LSTM Architecture

5 Implementation

5.1 Preprocessing

The project begins with the installation of libraries that include TensorFlow and Keras for model development, NLTK is included for the natural language tasks. The first part is the caption file and the corresponding image IDs to caption retrieval in order to effectively link captions to pictures during the training of the model.

Captions are stored in a dictionary along with its corresponding image ID. The captions are pre-processed by changing to lower case, removing special characters, adding start and end tokens. After cleaning captions, all the words are processed for tokenization. Keras Tokenizer is used to accomplish this, which encodes the words into an integer representation which is used for the model training vocabulary.

The dataset is divided into training and testing set. Then data generator function is used to generate forms of images during the model training in order to use memory wisely and training to be proper, so that the image and caption will act as a pair during the training of the CNN-LSTM models.

5.2 Developing Models

CNN-LSTM model with of the ResNet architecture is shown in Fig. 2. The image captioning model employs a CNN encoder and RNN decoder to generate captions for images. It employs

CNN as the encoder to obtain high-level feature representations from images, creating a vector with 2048 dimensions for each image. Then these features are sent through a densely connected layer to transform them down to the 256 needed to match the dimensions of the text embedding. At the same time, captions are also tokenized, and they are sent to an embedding layer which replaces them with a dense vector representation. To decrease the effect of overfitting, a dropout layer is applied to both image features and text embeddings.

The LSTM takes care of ignoring the padding tokens while processing the sequential text embeddings using a masking mechanism. This provides minimal noise by focusing on significant inputs. The image data and the output from the LSTM are combined using layers that retain the image and one of its partial captions. Such combined representation is further processed by fully connected layers and in the end layer, a prediction is made to which word comes next in the sentence according to the previous words. The training of the model is done using a cross-entropy loss function which makes the captions generated as similar as possible to the reference captions. The models' seamless combination of the visual features extracted by CNN and sequential features processed by LSTM allows for an effective success between interpreting images and generating captions that adequately describe the image.

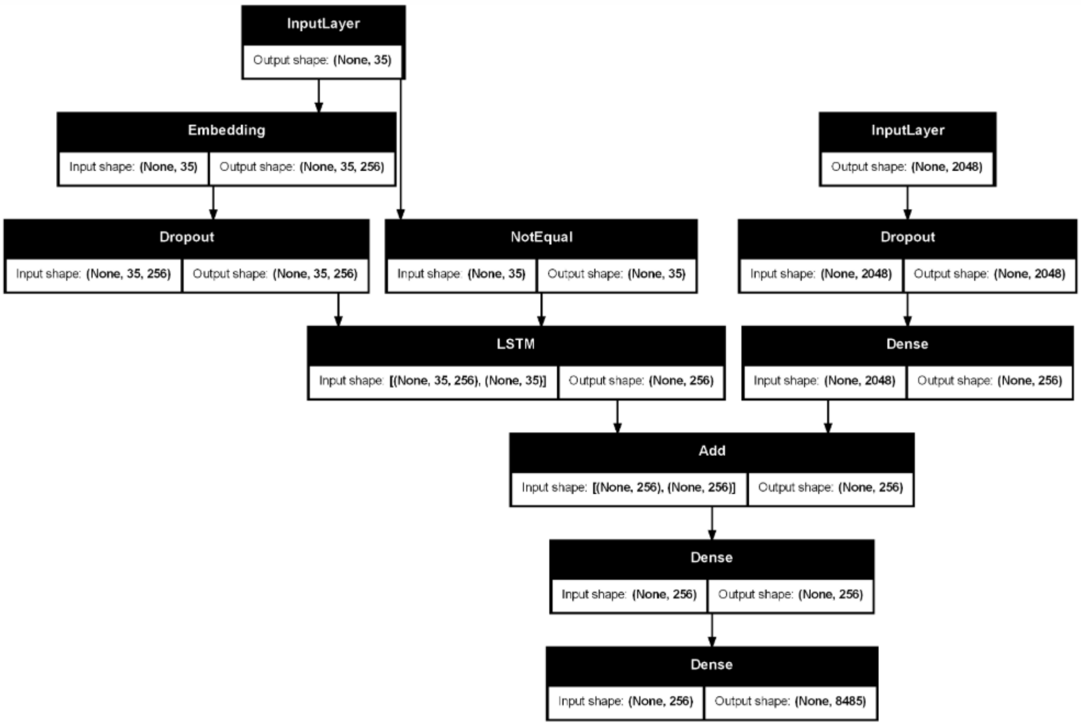

InputLayer
Output shape: (None, 35)
Embedding
Input shape: (None, 35)
Output shape: (None, 35, 256)
InputLayer
Output shape: (None, 2048)
Dropout
Input shape: (None, 35, 256)
Output shape: (None, 35, 256)
NotEqual
Input shape: (None, 35)
Output shape: (None, 35)
Dropout
Input shape: (None, 2048)
Output shape: (None, 2048)
LSTM
Input shape: [(None, 35, 256), (None, 35)]
Output shape: (None, 256)
Dense
Input shape: (None, 2048)
Output shape: (None, 256)
Add
Input shape: [(None, 256), (None, 256)]
Output shape: (None, 256)
Dense
Input shape: (None, 256)
Output shape: (None, 256)
Dense
Input shape: (None, 256)
Output shape: (None, 8485)

Figure 2: CNN (ResNet) - LSTM model

The architecture of the VGG16 and Xception models are followed in consistency with ResNet model. Pre-trained models of ResNet, VGG16 and Xception were used. The features extracted from VGG16 model is 4096 which is larger than ResNet and Xception which is 2048. The layers in CNN LSTM models are similar in all the three models. The model are trained for 5, 8 and 13 epochs to help the model to study the features and captions, also ensuring without overfitting.

Bagging is used by combining the predictions of multiple base models for the purposes of reducing variance and increasing reliability. VGG16, ResNet50, and Xception models, which all produce captions for the same set of images is used. For every image, the feature vector representing the image is given to each of the three models, and the respective captions they generate. The caption that is produced last is achieved by combining results, usually by taking the highest number of probable words. Comparing the collected information and how the models transform that information into captions, quality of the caption is evaluated using BLEU score. In this way, the strategy of bagging can be used to improve the captions generated.

Boosting applies an iterative process whereby hard-to-predict instances are focused on in this context, to improve the model performance. First, all images are given the same weight so that there is no bias. Once captions were constructed based on VGG16, ResNet50, and Xception, the weights were adjusted according to the correlation of the predictions and the BLEU scores of the predictions made. Moreover, images that recorded low BLEU scores meaning that the captioning was poor, are provided with increased weights in order to enhance their learning in the next iterations. Since the weights are easily adjusted when needed, it guarantees that the ensemble is able to capture the extracted features, and each caption is refined per iteration. The final captions created using the boosted ensemble method were measured using the BLEU scores to check the quality of caption. Boosting focuses on instances that are hard to learn, which can improve the final output from the individual models.

6 Evaluation and Results

This section discusses about the different experiments that were conducted for the research. The first experiment is conducted with VGG16 architecture of CNN model, which is also integrated with LSTM model. Consistency of CNN-LSTM model is maintained in all the experiments, only the architecture of CNN model was different in the first, second and third experiment. The second experiment is with Resnet50 architecture and the third one with Xception model. The fourth and fifth experiment was conducted to evaluate the models using ensemble techniques like bagging and boosting. The models of first three experiments are used for the ensemble methods. BLEU scores are used to evaluate the performance of the model, Meteor score is also considered for analysing the results.

6.1 Experiment 1 using VGG16 model

The VGG16 model comprises of pre-trained VGG16 model which is the encoder that extracts the image features of 4096 dimensions and an LSTM which decodes and produces the caption. The model achieved a BLEU-1 score of 0.5552 which shows that the model is actually able to capture the individual words. The model has BLEU-2 score of 0.3365, BLEU-3 score of 0.2347 and BLEU-4 score of 0.1312. The lower score of BLEU 2,3, and 4 suggests that the model is performing poor in generating captions considering the combination of words. An average Meteor score of 0.3978 was achieved that shows a fair degree of correlation between the actual captions and the predicted caption. The model was trained for 5,8 and 13 epochs. The best results were obtained by training the model for 8 epochs which is given above. The model will be overfitting after 8 epochs and further training was decreasing the performance of the model.

6.2 Experiment 2 using Resnet50 model

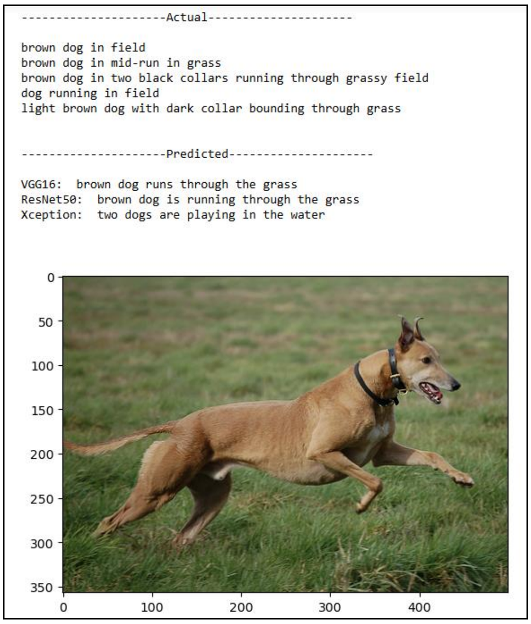
Compared to the VGG16 model, the ResNet50-based image captioning model achieved better results by scoring 0.5581 in the BLEU-1 test, thus depicting clear strength in mapping a single word with the one or more words of a given reference caption. Model achieved BLEU-2 (0.3394), BLEU-3 (0.2368), and BLEU-4 (0.1325) proving that ResNet50 had an advantage over the VGG16 in generating many words which are contextually relevant. The model was also able to score an average of 0.4089 in METEOR, indicating an increase in its efficiency of capturing the similarity and variations in the descriptions.

This model makes use of ResNet50, which is expected to improve its performance by functioning as the encoder and helps in creating more reliable 2048-Dimensional feature vectors of the input images. The features are used with a decoder that is integrated with LSTM and embedding for the text, helping derive the captions with meaning. The higher scores supports the argument that the ResNet50 excels in having a deeper architecture that enables better extraction of image features which improve the accuracy and contextual relevance of the captions.

6.3 Experiment 3 using Xception model

The Xception model is based on a deep and separable convolution method, which takes feature vectors of 2048 dimension and resize images to 299x299 which is supported by Xception's pre-trained network. The extracted features are forwarded to the fully connected layers while the text information is passed through the embedding and LSTM layers. The combination of these paths takes place in the decoder in order to create captions. One strength of the Xception model lies in its depth wise separable convolutions which make the model to be more discriminative while still computationally efficient. But considering the results of BLEU and METEOR score, the Xception model is performing poorly compared to the other two models.

The model was able to produce a BLEU-1 score of 0.4779 which indicates low performance in capturing the key individual words in the generated caption. The scores in BLEU-2: 0.2344, BLEU-3: 0.1473 and BLEU-4: 0.0701 indicates the inability of model to generate multiword phrases with contextual relevance. The Meteor score of 0.3180, which is the lowest among all models shows that the model is inefficient in both capturing semantic similarity and linguistic variability. Generated captions from all the models are given in Fig. 3.



```
---------------------Actual---------------------

brown dog in field
brown dog in mid-run in grass
brown dog in two black collars running through grassy field
dog running in field
light brown dog with dark collar bounding through grass


---------------------Predicted---------------------

VGG16:  brown dog runs through the grass
ResNet50:  brown dog is running through the grass
Xception:  two dogs are playing in the water
```

Figure 3: Caption generated using VGG16, ResNet50 and Xception models

6.4 Experiment 4 using Bagging Ensemble technique

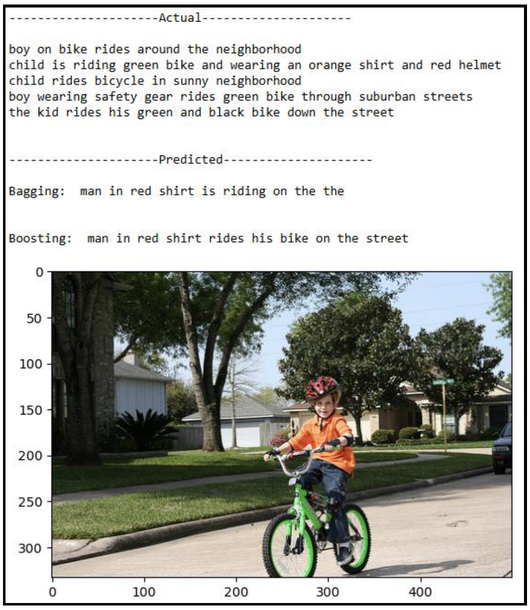
The performance of the ensemble technique that employs Bagging which integrates the predictions of VGG16, ResNet50 and Xception in the generation of image captions is said to be balanced. This is evidenced by the obtained BLEU-1 score of 0.4141 that confirms the model's ability to capture key individual words that pertain to the images. The obtained scores of BLEU-2 at 0.2051, BLEU-3 at 0.1146, and BLEU-4 at 0.0504 shows that the Bagging method is able to produce meaningful sequences of more than one word, but the performance deteriorates as longer sequences are produced. Such a decline in the performance in terms of the n-grams in BLEU score for n = 2, 3 and 4 can be expected since combining several models' predictions to form grammatically and contextually correct phrases is difficult. Baggin ensemble is seen to perform less comparing with the BLUE scores of the base models. This could be due because the different base models would be focusing and giving more emphasis on different words of the captions, which could be confusing for the model to capture the right words for the context and sometimes even generating repetitive words or grammatically incorrect captions.

Model achieved an average METEOR score of 0.2100. Even though the Bagging approach does not exceed performance metrics of every model separately on all metrics, it has the advantage of utilizing the VGG16, ResNet50 and Xception models which enhances the capability and variation of the captions. The captions predicted from this model have repetitive words and are sometimes unable to generate the sequence efficiently, this is seen as a drawback of the model. From this perspective, it is plausible to state that the ensemble strategy can be applied successfully when one wants to take advantage of the complementary capabilities of the models, even though it cannot really outperform each and every model all the time.

6.5 Experiment 5 using Boosting Ensemble technique

The ensemble technique that utilizes Boosting is unique in its way of optimizing the caption generation task as it alters the weight of the image samples through a performance-based optimization. The BLEU-1 score of 0.5552 means that the model does reasonably well in terms of grabbing the important key terms. This score is the best score achieved compared to all the models. The scores, BLEU-2: 0.3365, BLEU-3: 0.2347 and BLEU-4: 0.1312 indicate the same pattern of correlation with n-grams that are improving over the models based on images individually and are much higher compared to bagging. With the above behavior, Boosting makes it possible to take advantage of cumulative learning by manually activating the weak instances to create captioning that is more variable with the dataset.

Boosting's average METEOR score of 0.3978 is greater than that of Bagging. It indicates that this technique enhances the scope by placing greater emphasis on difficult samples rather than treating the collection as a whole. This result also shows how well Boosting performs in terms of the amount of variability present in a dataset. In general, the findings indicate that Boosting works well for producing image captions of good quality which makes it useful for tasks where the images are to be captioned in an iterative manner for improving the match with original caption. The predicted caption using Bagging and Boosting is shown in Figure 4, proving the efficiency of Boosting over Bagging in generating meaningful caption relevant to the image.



```
--------------------Actual--------------------

boy on bike rides around the neighborhood
child is riding green bike and wearing an orange shirt and red helmet
child rides bicycle in sunny neighborhood
boy wearing safety gear rides green bike through suburban streets
the kid rides his green and black bike down the street


--------------------Predicted--------------------

Bagging:  man in red shirt is riding on the the


Boosting:  man in red shirt rides his bike on the street
```

Figure 4: Caption generated using Bagging and Boosting

6.6 Discussion

The comparative approach towards the various models and techniques for image captioning is presented in this project. While the results appear to be very promising, there are quite a number of critical points that can be discussed about the results of the experiments as well as their enhancements.

Table 1: BLUE and METEOR scores of models

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
VGG16	0.5552	0.3365	0.2347	0.1312	0.3978
ResNet50	0.5580	0.3394	0.2368	0.1325	0.4089
Xception	0.4779	0.2344	0.1473	0.0701	0.3180
Boosting	0.5552	0.3365	0.2347	0.1312	0.3978
Bagging	0.4141	0.2051	0.1146	0.0504	0.2100

At first, while the VGG16, ResNet50 and Xception models were applied independently and to varying success, the high average BLEU scores for all VGG16 and ResNet50 show the models are performing well in capturing single word instances. The generated captions using Resnet50 and VGG16 are seen to be relevant and highly accurate in majority of the cases. Low BLEU scores for n-grams 2,3,4 imply inefficiency in capturing multi-word phrases. This could be a consequence of the weakness of the CNN-LSTM architecture employed which essentially makes use of only a fixed number of pre-trained CNN models and a single layer LSTM for decoding purposes. Previous works acknowledged the relevance of attention mechanisms for improved understanding of context and more informative captioning (Wu et al., 2023). Not using such a mechanism in the present study could account for the lower scores experienced more so in the case of BLEU-2, BLEU-3 and BLEU-4 where longer n-grams were compared. Where attention mechanisms or transformer-based architectures are used, they allow the model during the decoding to concentrate on most relevant image parts which would minimize context loss in the produced caption.

Secondly, the Boosting ensemble methods proved to be effective in combining multiple models so as to improve image captioning performance. Boosting gave quite an advantage as it made use of the diversity of the base models, and its METEOR score is also better. Bagging method is showing less value for BLEU and METEOR scores and the generated captions are out of context in many case. Sometimes the captions generated by Bagging are grammatically incorrect, as different model are giving more emphasis of different words, hence the model was struggling to capture the exact word. On the other hand, the Boosting method achieved better results with regards to the context of the image. Boosting is producing results and the score are similar or identical to VGG16 model, which could be due to updating of weights on the performance of the individual models. As the average scores of VGG16 and ResNet are similar and there are minimal amount of data, it might be difficult for the ensemble to capture the difference in weights and resulting a model to dominate. The major drawback in the ensemble methods was the simple use of the weighting technique in Boosting, where all the samples were given the same weights regardless of how difficult the caption was. It may be beneficial to incorporate adaptive weights into its iterations. Additionally, the methods of aggregation in Bagging could have been more advanced through the use of weighted voting that explicitly takes each base model’s contribution.

The model was trained for 5,8 and 13 epochs and noted that increasing the training epochs resulted in overfitting. The best results were obtained by training the model for 8 epochs. This indicates that the dataset used was not large or diverse enough to take advantage of the functions offered by deep learning models. Furthermore, the analysis was dependent on BLEU and METEOR scores which, although widely reported, do not always match well with the human assessment of the quality of a caption. Adding more human-centered metrics would be ideal in this case as they would allow a better assessment of the models' performance. The study proved the advantage of combining models using ensemble techniques, this could be justified within the existing literature. It has been shown previously that ensemble techniques perform better when the base models have diverse strengths. However, in this study, three different CNN architectures were used but the complementary aspects of these models were not closely investigated. Deep understanding on the base models' strength and weakness in the image extraction would help to integrate the ensemble techniques to improve its performance.

7 Conclusion and Future Work

The research question was "How an ensemble of different deep learning models like Convolutional Neural Network, Long Short-Term Memory and its different architectures can be effectively combined to enhance the quality and accuracy of generated image captions?". To address this, we considered the different CNN-LSTM models with architectures VGG16, ResNet50 and Xception models, examined them for the use of ensemble techniques, specifically Bagging and Boosting. The aim of these objectives was to evaluate the individual performance of the different models, point out the limitations associated with each of them, and assess whether the ensemble models can improve the performance and caption quality.

This work has been able to put together and test practically the approaches and significance of the image captioning task was accomplished. ResNet50 being the best individual model outperformed all others in the tasks by achieving the highest score. ResNet50 achieved better BLEU and METEOR score than VGG16 and Xception, also geenrating captions that are able to provide context. While Bagging was unable to capture and balance the contributions of different models, captions with irrelevant words and poor grammar structures were generated in some cases. On the other hand, Boosting enabled enhancement of the captions produced for the images in various iterations followed by optimal BLEU and METEOR scores. Thus, it is justified that the Boosting as other ensemble methods do help in mitigating the problems that are encountered with individual models and can be used for enhancing the quality of captions.

Even though the ensemble models were performing good, there are some drawbacks. The ability of the ensemble methods to focus on the strengths and weakness of the individual models could have been explored deeply to improve the performance. The distribution of weights to be considered in the ensemble methods should be able to capture the capability of the base models. The size of the dataset would probably have been too small to adequately train and examine the models' capacities as shown by the results which indicated overfitting as well. BLEU and METEOR scores were the two principal measures, these helped to analyze the quality of captions, but do not seem to provide enough information on how human beings judge the context of captions quality.

For future work, attention mechanism can be incorporated with the model to improve the quality of captions by capturing the important parts of an image. Different evaluation metric like SPICE can be used to evaluate the model, which would be able to judge the captions generated by the model based on the human judgements (Anderson et. al. 2016). There are larger dataset like Flickr30k which can be used to train the model, with a system having more computational power, the model can be trained with larger dataset and extensive testing can be carried out. A system with audio descriptions can be built to describe the generated captions which can help people with visual impairment.

References

Alzubi, J.A., Jain, R., Nagrath, P., Satapathy, S., Taneja, S. and Gupta, P. (2020). Deep image captioning using an ensemble of CNN and LSTM based deep neural networks. *Journal of Intelligent & Fuzzy Systems*, pp.1–9. doi:https://doi.org/10.3233/jifs-189415.

Anderson, P., Fernando, B., Johnson, M. and Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14* (pp. 382-398). Springer International Publishing. doi: https://doi.org/10.1007/978-3-319-46454-1_24

Ardabili, S., Mosavi, A. and Várkonyi-Kóczy, A.R. (2020). Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods. *Lecture Notes in Networks and Systems*, pp.215–227. doi:https://doi.org/10.1007/978-3-030-36841-8_21.

Berger, U., Stanovsky, G., Abend, O. and Frermann, L., (2024). Surveying the Landscape of Image Captioning Evaluation: A Comprehensive Taxonomy and Novel Ensemble Method. arXiv preprint arXiv:2408.04909. https://arxiv.org/abs/2408.04909

Bhatt, C., Rai, S., Chauhan, R., Dua, D., Kumar, M. and Sanjay Kumar Sharma (2023). Deep Fusion: A CNN-LSTM Image Caption Generator for Enhanced Visual Understanding. doi:https://doi.org/10.1109/cisct57197.2023.10351389.

Cai, H., Lin, J., Lin, Y., Liu, Z., Tang, H., Wang, H., Zhu, L. and Han, S. (2022). Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications. *ACM Transactions on Design Automation of Electronic Systems*, 27(3), pp.1–50. doi:https://doi.org/10.1145/3486618.

Carlos, J., Heber López-Osorio, Mateo Rico-García and Hermes Fandiño-Toro (2024). Deep learning as a powerful tool in digital photoelasticity: Developments, challenges, and implementation. *Optics and Lasers in Engineering*, 180, pp.108274–108274. doi:https://doi.org/10.1016/j.optlaseng.2024.108274.

Chauhan, S. and Daniel, P. (2022). A Comprehensive Survey on Various Fully Automatic Machine Translation Evaluation Metrics. *Neural Processing Letters*. doi:https://doi.org/10.1007/s11063-022-10835-4.

Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [online] doi:https://doi.org/10.1109/cvpr.2017.195.

Datta, G., Joshi, N. and Gupta, K. (2022). Analysis of Automatic Evaluation Metric on Low-Resourced Language: BERTScore vs BLEU Score. *Lecture Notes in Computer Science*, pp.155–162. doi:https://doi.org/10.1007/978-3-031-20980-2_14.

Dong, X., Qian, L. and Huang, L. (2017). A CNN based bagging learning approach to short-term load forecasting in smart grid. doi:https://doi.org/10.1109/uic-atc.2017.8397649.

Fan, W. and Zhang, K. (2009). Bagging. *Encyclopedia of Database Systems*, pp.206–210. doi:https://doi.org/10.1007/978-0-387-39940-9_567.

Fudholi, D.H., Zahra, A. and Nayoan, R.A.N. (2022). A Study on Visual Understanding Image Captioning using Different Word Embeddings and CNN-Based Feature Extractions. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*. doi:https://doi.org/10.22219/kinetik.v7i1.1394.

Ganaie, M.A., Hu, M., Malik, A.K., Tanveer, M. and Suganthan, P.N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115, p.105151. doi:https://doi.org/10.1016/j.engappai.2022.105151.

Gkelios, S., Sophokleous, A., Plakias, S., Boutalis, Y. and Chatzichristofis, S.A. (2021). Deep convolutional features for image retrieval. *Expert Systems with Applications*, 177, p.114940. doi:https://doi.org/10.1016/j.eswa.2021.114940.

Islam, Md.Z., Islam, Md.M. and Asraf, A. (2020). A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in Medicine Unlocked*, 20, p.100412. doi:https://doi.org/10.1016/j.imu.2020.100412.

Jain, G., Sharma, M. and Agarwal, B. (2019). Spam detection in social media using convolutional and long short term memory neural network. *Annals of Mathematics and Artificial Intelligence*, 85(1), pp.21–44. doi:https://doi.org/10.1007/s10472-018-9612-z.

Jinsakul, N., Tsai, C.-F., Tsai, C.-E. and Wu, P. (2019). Enhancement of Deep Learning in Image Classification Performance Using Xception with the Swish Activation Function for Colorectal Polyp Preliminary Screening. *Mathematics*, 7(12), p.1170. doi:https://doi.org/10.3390/math7121170.

Khandelwal, R. (2020). *BLEU — Bilingual Evaluation Understudy*. [online] Medium. Available at: https://towardsdatascience.com/bleu-bilingual-evaluation-understudy-2b4eab9bcfd1 [Accessed 5 May 2022].

Kotu, V. and Deshpande, B. (2019). Data Science Process. *Data Science*, pp.19–37. doi:https://doi.org/10.1016/b978-0-12-814761-0.00002-2.

Kuo, C.C.J. (2016). Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, 41, pp.406-413. doi: https://doi.org/10.1016/j.jvcir.2016.11.003

Lee, S., Lee, J., Moon, H., Park, C., Seo, J., Eo, S., Koo, S. and Lim, H. (2023). A Survey on Evaluation Metrics for Machine Translation. *Mathematics*, [online] 11(4), p.1006. doi:https://doi.org/10.3390/math11041006.

Sai, A.B., Mohankumar, A.K. and Khapra, M.M. (2023). A Survey of Evaluation Metrics Used for NLG Systems. *ACM Computing Surveys*, 55(2), pp.1–39. doi:https://doi.org/10.1145/3485766.

Santi, D., Amil Ahmad Ilham, None Syafaruddin and Nurtanio, I. (2024). Image Caption Generation Through the Integration of CNN-Based Residual Network Architectures and LSTM. 4, pp.227–232. doi:https://doi.org/10.1109/icicos62600.2024.10636926.

Sharma, D., Dhiman, C. and Kumar, D. (2023). Evolution of visual data captioning Methods, Datasets, and evaluation Metrics: A comprehensive survey. *Expert Systems with Applications*, 221, p.119773. doi:https://doi.org/10.1016/j.eswa.2023.119773.

Sharma, S., Guleria, K., Tiwari, S. and Kumar, S. (2022). A deep learning based convolutional neural network model with VGG16 feature extractor for the detection of Alzheimer Disease using MRI scans. *Measurement: Sensors*, 24, p.100506. doi:https://doi.org/10.1016/j.measen.2022.100506.

Sharma, V. and Singh, N. (2021). Deep Convolutional Neural Network with ResNet-50 Learning algorithm for Copy-Move Forgery Detection. *2021 7th International Conference on Signal Processing and Communication (ICSC)*. doi:https://doi.org/10.1109/icsc53193.2021.9673422.

Stefanini, M., Cornia, M., Baraldi, L., Cascianelli, S., Fiameni, G. and Cucchiara, R. (2022). From Show to Tell: A Survey on Deep Learning-based Image Captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.1–1. doi:https://doi.org/10.1109/tpami.2022.3148210.

Vedantam, R., Tech, V., Zitnick, C. and Parikh, D. (2015). *CIDEr: Consensus-based Image Description Evaluation*. [online] Available at: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Vedantam_CIDEr_Consensus-Based_Image_2015_CVPR_paper.pdf.

Wu, J.-S., Liu, Y., Ge, F. and Yu, D.-J. (2023). Prediction of protein-ATP binding residues using multi-view feature learning via contextual-based co-attention network. *Computers in Biology and Medicine*, 172, pp.108227–108227. doi:https://doi.org/10.1016/j.compbiomed.2024.108227.

Zhang, C., Iqbal, I., Bhatti, U.A., Liu, J. and Sarhan, N. (2024). ResNet50 in remote sensing and agriculture: evaluating image captioning performance for high spectral data. *Environmental Earth Sciences*, [online] 83(23). doi:https://doi.org/10.1007/s12665-024-11950-2.

Zhang, K., Li, P. and Wang, J. (2024). A Review of Deep Learning-Based Remote Sensing Image Caption: Methods, Models, Comparisons and Future Directions. *Remote Sensing*, 16(21), pp.4113–4113. doi:https://doi.org/10.3390/rs16214113.

Zhou, Q., Qu, Z., Guo, S., Luo, B., Guo, J., Xu, Z. and Akerkar, R. (2021). On-Device Learning Systems for Edge Intelligence: A Software and Hardware Synergy

Perspective. *IEEE Internet of Things Journal*, 8(15), pp.11916–11934. doi:https://doi.org/10.1109/jiot.2021.3063147.