Comparing the Performance effect of Automatic Image Caption Generation Models Builed by Using Various Convolutional Architectures

Venkata Satya Sai Ajay Daliparthi
Department of computer science
Blekinge Tekniska Högskola
Karlskrona, Sweden
veda18@student.bth.se

***Abstract*—Automatically generating a natural language description of an image has attracted interests recently both because of its importance in practical applications and because it connects two major artificial intelligence fields: computer vision and natural language processing. Deep Neural Networks with Encoder-Decoder frameworks are mostly used in dealing with these Tasks. In this paper, we compared the performance effect of four Different CNN Architectures VGG16, DenseNet121, MobileNet, and ResNet50 (Pre-trained on ImageNet dataset) in Image captioning. The Bilingual Evaluation Understudy (BLEU) metric is used to evaluate the results generated by four models . The mean BLEU scores for the Models are VGG16(0.015), DenseNet121(0.010), MobileNet(0.013) and ResNet50(0.024). We evaluated the models on Flickr8K Dataset. Experimental results show that using ResNet50 as CNN encoder shows a huge difference in performance compared to more recent State-of-the-art image classification Networks like DenseNet121 and MobileNet.**

***Index Terms*—Image Captioning, Deep neural networks, Multimodal embedding, Encoder–decoder framework, Image classification, Natural Language Processing, Computer Vision**

I. Introduction

It is natural for humans to relatively easily describe the surrounding environments they are in. Being able to automatically describe the content of an image using properly formed English sentences is a very challenging task. Although great progress has been made in various computer vision tasks, such as object recognition [1], attribute classification [2], action classification [3], image classification [4], and scene recognition [5]. Indeed, a description must capture not only the objects contained in an image, but it also must express how these objects relate to each other as well as their attributes and the activities they are involved in.

Using a computer to automatically generate a natural language description for an image, which is defined as image captioning (shown in fig.1), is connecting both research communities of computer vision and natural language processing. Since much of human communication depends on natural language enabling computers to describe the visual world will lead to a great number of possible applications, such as producing natural human-robot interactions, early childhood education, information retrieval, and visually impaired assistance, and so on.

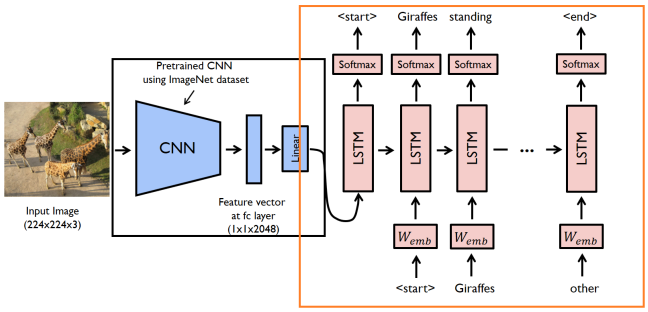


Fig. 1. Flow of Image Captioning process

In Machine Translation by using Recurrent Neural Networks (RNNs) [6], an "encoder" RNN reads the source sentence and transforms it into a rich fixed-length vector representation, which in turn in used as the initial hidden state of a "decoder" RNN that generates the target sentence. image captioning can be formulated as a translation problem, where the input is an image, while the output is a sentence [7]. The main inspiration for our work comes from this method by replacing decoder RNN with a deep convolution neural network (CNN). During the past few years, it has been convincingly shown that CNNs can produce a rich representation of the images [8] [9] [10] [11]. By embedding the image to a fixed-length vector, which is used as input to Decoder RNN that generates a natural language description of the Image. With the fast development of deep neural networks, employing more powerful network structures as language models and/or visual models will undoubtedly improve the performance of image description generation [12].

In this work, we compared the performance effect and training time of various Convolutional Architectures in the Image captioning system. By, building four different Image captioning systems by using four different Convolutional

Architectures like VGG16 [9], DenseNet121 [8], MobileNet [11], and ResNet50 [10] as CNN Encoders. Long Term Short Memory Recurrent Neural Network [13] is used as RNN Decoder for all four Image captioning systems.
All four models are evaluated on Flickr8k Dataset and Used Bilingual Evaluation Understudy (BLEU) metric. The BLEU metric scores for four Models are shown as VGG16(0.015), DenseNet121(0.010), MobileNet(0.013) and ResNet50(0.024). By observing the Experimental results it is found that using ResNet50 as CNN encoder shows a huge difference in performance compared to more recent State-of-the-art image classification Networks like DenseNet121 and MobileNet. The Training Time for both VGG16 and MobileNet is 4.3 min which is lower compared to DenseNet121 and ResNet50 having training time of 4.7min and 4.47 minutes respectively. All the above results are obtained after Training each model for 5 epochs.

II. RELATED WORK

Shuang Bai and Shan [12] presented all the Traditional and deep learning based image captioning methods In a survey on automatic image caption generation. Traditional methods for solving this task mainly include Retrieval and template-based methods. Given a query image, retrieval-based methods produce a caption for it through retrieving one or a set of sentences from a pre-specified sentence pool. The generated caption can either be a sentence that has already existed or a sentence composed from the retrieved ones.
Hodosh et al. [14] framed image captioning as a ranking task, then employed the Kernel Canonical Correlation Analysis technique to project image and text items into a common space, where training images and their corresponding captions are maximally correlated. Kuznetsova et al. [15] proposed a tree-based method to compose image descriptions by making use of captioned web images. After performing image retrieval and phrase ex- traction, the authors took extracted phrases as tree fragments and model description composition as a constraint optimization problem.
In template-based methods, image captions are generated through a syntactically and semantically constrained process. A specified set of visual concepts needs to be detected first in order to use a template-based method. Then those visual concepts are connected through sentence templates or specific language grammar rules or combinatorial optimization algorithms to generate captions.
Yang et al. [16] proposed a sentence template for generating image descriptions, where a quadruplet (Nouns-Verbs-Scenes-Prepositions) is utilized as a sentence template. They used detection algorithms to estimate objects and scenes in this image. Then employed a language model trained over the Gigaword corpus3 to predicate verbs, scenes, and prepositions that may be used to compose the sentence. Mitchell et al. [17] employed computer vision algorithms to process an image and represent this image by using ⟨objects, actions, spatial relationships⟩ triplets. After that, they formulated the image description as a tree-generating process based on the visual recognition results.
Due to the recent advancements in Deep learning, recent work begins to rely on deep neural networks for automatic image captioning. With inspiration from retrieval-based methods, researchers propose to utilize deep models to formulate image captioning as a multi-modality embedding and ranking problem.
Karpathy et al. [18] proposed a model in which they embedded the sentence fragments and image fragments into a common space for ranking sentences for a query image. They use dependency tree relations of a sentence as sentence fragments and use detection results of the Convolutional Neural Network method. Yan and Mikolajczyk [19] proposed to use deep Canonical Correlation Analysis to match images and sentences. They used a deep Convolutional Neural Network to extract visual features from images and use a stacked network to extract textual features from Frequency-Inverse Document Frequency represented sentences.
In multimodal learning-based image captioning methods image features are first extracted by using a feature extractor, such as deep convolutional neural networks. Then, obtained image features are forwarded to a neural language model, which maps the image feature into the common space. Schuster and Paliwal [20] presented an approach to align image regions represented by a Convolutional Neural Network and sentence segments represented by a Bidirectional Recurrent Neural Network to learn a multimodal Recurrent Neural Network model to generate descriptions for image regions.
The encoder-Decoder framework used in Machine Translation is adopted to generate captions for images. Kiros et al. [7] introduced the encoder-decoder framework into image captioning research to combine joint image-text embedding models and multimodal neural language models, so that given an image input, a sentence output can be generated word by word like language translation. They used LSTM RNN's to encode textual data and a deep CNN's to encode visual data. Vinyals et al. [21] used a deep Convolutional Neural Network as an encoder to encode images and use Long Short-Term Memory (LSTM) RNN's to decode obtained image features into sentences. Donahue et al. [22] also adopted a deep Convolutional Neural Network for encoding and Long Short-Term Memory Recurrent Networks for decoding to generate a sentence description for an input image. Pu et al [23] proposed a semi-supervised learning method under the encoder-decoder framework to use a deep Convolutional Neural Network to encode images and a Deep Generative Deconvolutional Network to decode latent image features for image captioning.

The selection of better convolutional architectures for encoding image and Recurrent architecture for Decoding image vector results in the improvement of performance in the image captioning system [12] .
Karen Simonyan Andrew Zisserman [9] proposed a deep neural network model for image classification of increasing depth using an architecture with very small (3×3) convolution

filters, which shows that a significant improvement on the prior-art configurations

Kaiming He et al. [10] introduced a residual learning framework to ease the training of networks that are substantially deeper than those used previously. They provided comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth.

Andrew G. Howard et al. [11] introduced an efficient model that uses depth-wise separable convolutions to build light weight deep neural networks. The proposed model has shown effectiveness across a wide range of applications and uses cases including object detection, fine-grain classification, face attributes, and large scale geo-localization

Gao Huang et al. [8] introduced the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. DenseNets obtained a significant improvement over the state-of-the-art networks (like VGG19, ResNets, and Googlenet)on most of them while requiring less cost of computation to achieve high performance.

Long Term Short Memory (LSTM)'s [13] is used as RNN Decoder to generate text sequences. Because Gated recurrent units are proved to be computationally effective than LSTM's. But, LSTM's are proved to be computationally effective than GRU's. So, LSTM's are chosen as RNN Decoder for our Image captioning system.

III. Background

A. *Image Classification*

In computer vision, Image classification is an approach of classification based on contextual information in images (shown in figure.2 below). Convolutional networks (ConvNets) have recently enjoyed a great success in the large-scale image and video recognition. which has become possible due to the large public image repositories, such as ImageNet, and high-performance computing systems, such as GPUs. Many different architectures that are proposed to deal with this task have applications in various computer vision tasks like semantic segmentation, object detection, human pose estimation, and 3d reconstruction, etc. The four different Convolutional

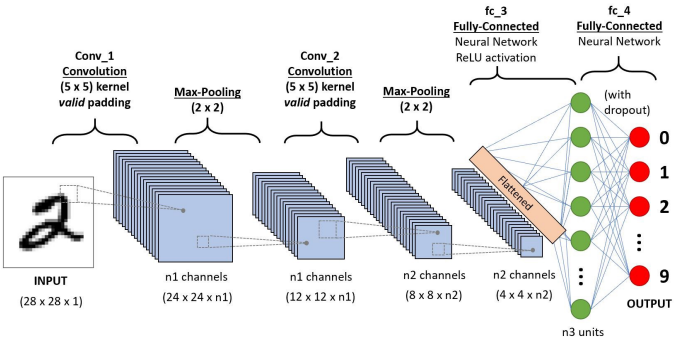


Fig. 2. Convolution sequence to classify hand written digits

architectures used in this work are as follows:

*1) **VGG16**:* The VGG16 Architecture [9] investigated the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. The input to VGG16 architectures ConvNets is a fixed-size 224 × 224 RGB images. A linear transformation of the input channels is done by passing the image through a stack of convolutional (conv.) layers, with a very small receptive field: 3 × 3. The convolution stride is fixed to 1 pixel; the spatial padding of Conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1 pixel for 3 × 3 Conv. layers. Spatial pooling is done by five Max-pooling layers. Max-pooling is done over a 2 × 2 pixel window, with stride 2. Finally, these stack of convolutional layers are followed by Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000- way ILSVRC classification and thus contains 1000 channels (one for each class).(as shown in figure 3 below) This architecture provided state-of-the-art results in the localization and classification tracks respectively.

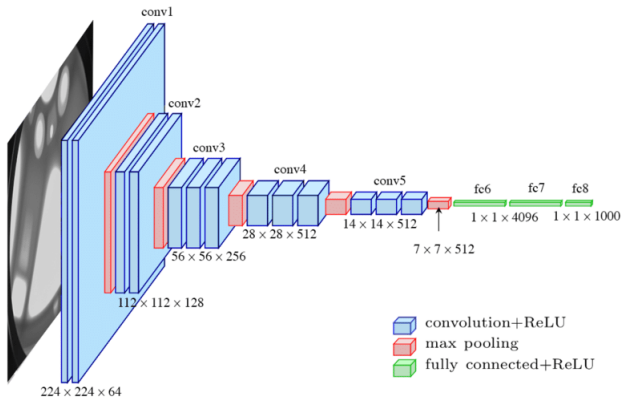


Fig. 3. Very deep convolutional networks for large-scale image recognition

*2) **ResNet**:* ResNet [10] introduced the residual learning framework to ease the training of networks that are substantially deeper than those used previously. The residual learning block (shown in figure 4 below) is the main contribution of ResNet, which allowed gradient flow. ResNet architecture is mostly inspired by VGG networks. The convolutional layers mostly have 3×3 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled to preserve the time complexity per layer. The downsampling is performed directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax. They adopted Batch Normalization right after each convolution and before activation. Residual nets achieved a 3.57 percent error on the ImageNet test set. It also has applications in various computer vision tasks like object detection, localization, and

segmentation.

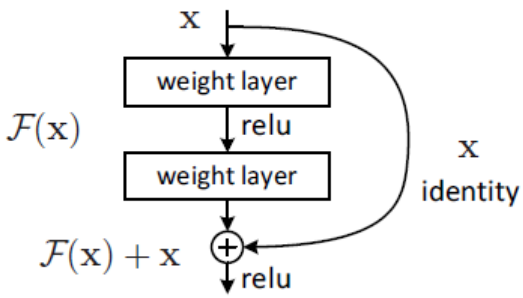


Fig. 4. Residual learning: a building block.

*3) **MobileNet**:* MobileNets [11] are a class of efficient models called MobileNets for mobile and embedded vision applications. They introduced two simple global hyperparameters that efficiently trade-off between latency and accuracy. These hyper-parameters allowed the model builder to choose the right sized model for their application based on the constraints of the problem. The MobileNet architecture is based on depthwise separable convolutions which are a form of factorized convolutions that factorize a standard convolution into a depthwise convolution and a 1 × 1 convolution called a pointwise convolution(as shown in figure 5 below). The standard convolutional layer is parameterized by convolution kernel K of size DK × DK× M × N where DK is the spatial dimension of the kernel assumed to be square and M is the number of input channels and N is the number of output channels as defined previously. MobileNet has can be applied to various recognition tasks like object detection, Landmark Recognition, Face Attributes, and fine-grain Classification, for efficient on-device intelligence.

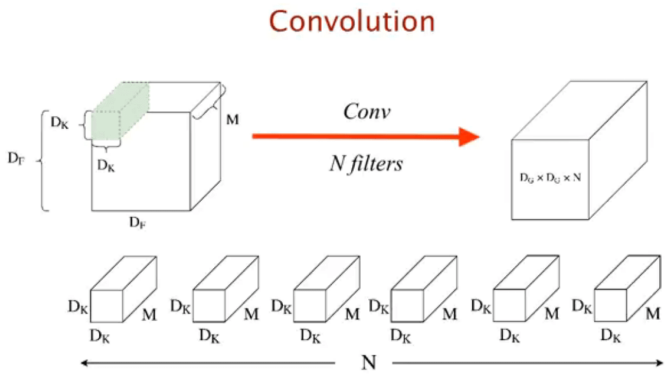


Fig. 5. Depthwise Separable Convolution

*4) **DenseNet**:* Dense Convolutional Network (DenseNet) [8] connects each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections—one between each layer and its subsequent convolutional layers. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. Consider a single image x that is passed through a convolutional network. The network comprises L layers, each of which implements a non-linear transformation H(x). H can be a composite function of operations such as Batch Normalization (BN), rectified linear units (ReLU), Pooling, or Convolution (Conv). The DenseNet architecture consists of many DenseBlocks(shown in figure 6 below) which are followed by a linear layer. DenseNet obtained significant improvements over the state-of-the-art on most of them, whilst requiring less computation to achieve high performance .

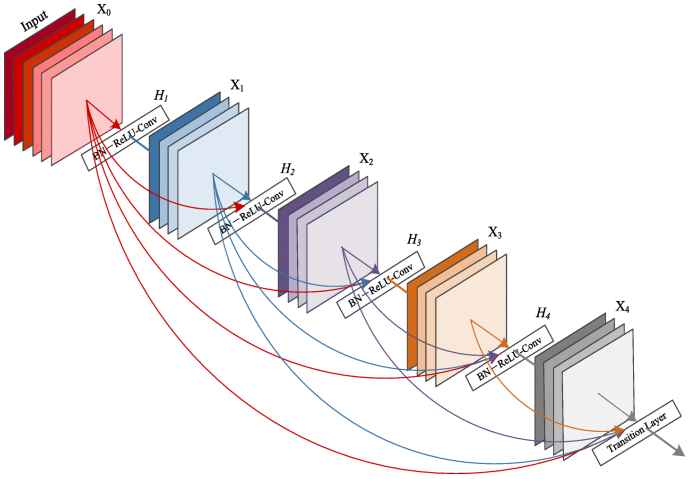


Fig. 6. A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input.

B. *Language Models*

A statistical language model is a likelihood probability distribution over groupings of words. Given such a sequence, say of length m, it assigns a probability to the whole sequence. Recently, neural-network-based language models have demonstrated better performance than classical methods. It has many applications such as Optical Character Recognition, Handwriting Recognition, Machine Translation, Spelling Correction, Image Captioning, Text Summarization, and much more. Recurrent neural networks and then networks with a long-term memory like the Long Short-Term Memory network, or LSTM are mostly used in dealing with sequential data. They allow the models to learn the relevant context over much longer input sequences than the simpler feed-forward networks.

*1) **Long Short-Term Memory (LSTM)**:* Long Short-Term Memory networks also called LSTMs [13] are special kinds of RNNs that are capable of learning long-term dependencies. They are designed to overcome the long-term dependency problem by Remembering information for long periods of time. LSTM has this chain-like structure (shown in figure

7 below) with four different operations. They are sigmoid, Tanh, pointwise multiplication(X), and pointwise addition(+). The network has three different gates that regulate information flow in an LSTM cell. A forget gate, input gate, and output gate. Forget gate decides what information should be kept

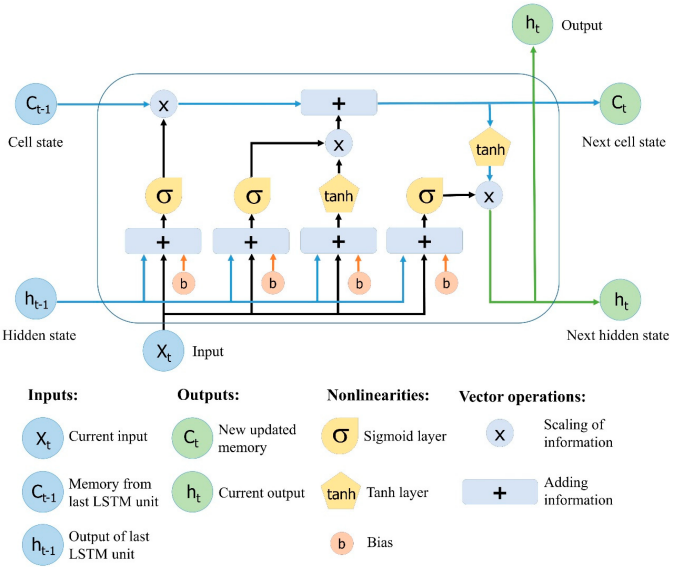


Fig. 7. LSTM cell and it's operations

or thrown away. The information from the previous hidden state and current input is passed through the sigmoid function. Through the input gate we pass the information from hidden state and current input into both sigmoid function and Tanh function, then multiply the sigmoid output with Tanh output. At the cell state gate, all the information need to calculate cell state should present. First, the cell state is pointwise multiplied by forget vector and then we take the output from the input gate and do a pointwise addition. This gives the new cell state. At last, we have an output gate, which decides what the next hidden state should be. LSTM's are used as RNN Decoders in our work.

IV. METHOD

To evaluate the performance of four different CNN architectures on Image captioning, we need to build four different Image captioning models with each CNN architecture individually. After building the models and generating the results, we will compare those models based on the selected performance metrics. Thus Experiment is chosen as a research methodology to perform this study.

Aim: The aim of this experiment is to compare the performance effect of the four different CNN architectures in the Automatic Image captioning models.

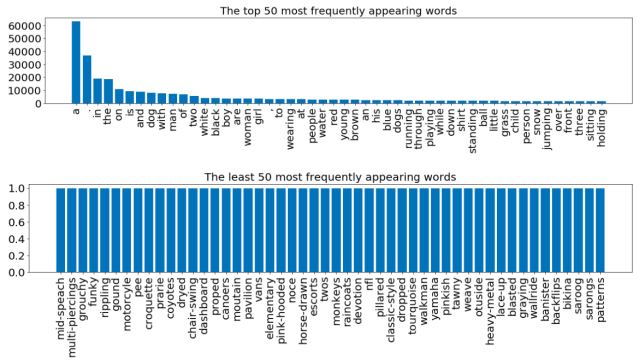
Null Hypothesis (Ho): "The Automatic Image captioning models built by using four different CNN Architectures perform similarly"

Alternative Hypothesis (Ha): "The Automatic Image captioning models built by using four different CNN Architectures doesn't perform similarly"

A. *Data Set*

We used Flickr8K Dataset for this task because It is small in size. So, the model can be trained easily. For each image in the data set 5 captions are provided. Flickr8K dataset Contains a total of 8092 images in JPEG format with different shapes and sizes. Of these 6000 are used for training, 1000 for test and 1000 for development. It also contains text files describing train set, testset. Flickr8k.token.txt contains 5 captions for each image i.e. total 40460 captions.

By performing basic data analysis on the dataset, The top 50 most frequently appearing words and the least 50 most frequently appearing words are presented in figure 9 below.



B. *Data Pre-Processing*

Data Pre-processing is done for both the images and textual data before feeding them into the neural network model. This improves the performance of the results generated by the models.

1) ***Images****:* Every image in the dataset is resized to 224 X 244 and normalized in order to make it easier to feed into neural networks. Also, images are passed into generators which takes all the images as input and results in batches of images as output which are used in training the neural network by mini-batches.

2) ***Image captions****:* All the image captions are lowercased. Any numerical or punctuations and symbolic characters are removed. neural networks cannot process the textual data so, all the textual data is Tokenized using keras.preprocessing.text method.

C. *Vocabulary*

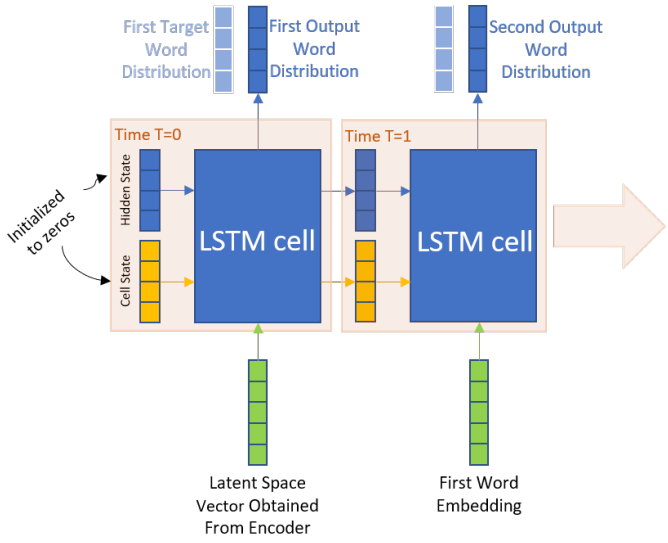
sequence models do not understand the symbolic language. vocabulary in language models refers to embedding every word in a higher dimensional real number space with which we can operate to handle the recurrent neural network. Embeddings are also useful in other natural language processing applications as they allow the practitioner to examine the word or character manifold once it is mapped to a 2-dimensional space, typically using the T-SNE algorithm. we want a vocabulary that is both expressive and as small as possible. A smaller vocabulary will result in a smaller model that will train faster.

D. Model Architecture

The Automatic image Captioning model in our work follows an Encoder-Decoder Architecture that communicate through a latent space vector. we will map an image to some intractable latent space via encoding and map the latent space representation of the image to the sentence space via decoding.

1) ***Encoder****:* The convolutional neural networks are used to encode images into latent space vectors. Here we took four different convolutional neural network architectures VGG16, DenseNet121, MobileNet, and ResNet50 and built four models individually. Here we performed Transfer Learning by importing the CNN architectures pre-trained on ImageNetdataset. Transfer Learning can improve the results of the model [24]. Then by removing the last layer (FC1000) of CNN architectures we will tune the model to fit into our task. Finally, 1024-dimensional vector in the latent space, which we will feed as the first input to our LSTM model (at time t=0).

2) ***Decoder****:* The goal of the Decoder is to optimize to find the right set of weights to accommodate the whole dictionary of words (characters in char-to-char models). This means that for every word in our sentence (which is a sequence) we are going to feed the word as input and get some output which is typically a probability distribution over the whole dictionary of words. (shown in figure below) This way we can obtain the word that the model thinks fits the most given the previous word. we are going to use a single layer LSTM to map the latent space vector to the word space. By using a fully connected layer between the hidden state space and the vocabulary space, mapping from the hidden state space to the vocabulary (dictionary) space is achived. The key idea here is



to feed the latent space vector that represents the image as the input to the LSTM cell at time t=0. Beginning at time t=1 we can start feeding our embedded target sentence into the LSTM cell as a part of the teacher forcer algorithm.

E. Training

The training of the four neural network architectures (VGG16, DenseNet121, MobileNet, and ResNet50) were performed on the GPU with Keras 2.3.0 and TensorFlow 1.12.02 as the back-end. ”categorical cross-entropy” Loss function and ”Adam” optimizer is used to calculate the losss and optimize the gradient descent respectively. Batch normalization and Dropout are used in some models. These parameter configurations are most commonly used parameters for Image Captioning Task [18] [19] [7] [21]. The Experimental settings of our work is shown below:

- CPU: Intel Xeon E5-1620 v4 3.50 GHz
- GPU: Nvidia Gefore GTX 1080, 8 GB VRAM
- RAM: 16 GB DDR4

F. Evaluation metrics

Being plagued by the complexity of the outputs, image captioning methods are difficult to evaluate. In order to compare image captioning systems as for their capability to generate human-like sentences with respect to linguistic quality and semantic correctness. The commonly used automatic evaluation metrics include BLEU, ROUGE-L, METEOR and CIDEr [12]. BLEU, ROUGE-L and METEOR are originally designed to judge the quality of machine translation.

Bilingual Evaluation Understudy (BLEU):
BLEU proposed by kishore papineni et al [25] is used for evaluating the quality of the machine-translated text. We can use BLEU to check the quality of our generated caption. BLEU is selected as evaluation metric among others because, It is language-independent, Easy to understand, and easy to compute. Its value lies between [0,1]. Higher the score better the quality of the caption. we calculate the BLEU scores in Python by using the NLTK library for sentences and documents.

V. RESULTS

After Training the models and generating the results, all the generated captions are evaluated by using BLEU scores and Time taken for training the models are observed. The performance results are presented as below:

A. Performance

Model	Mean BLEU	Time
VGG16	0.015	**4.3 min**
DenseNet	0.010	4.7 min
MobileNet	0.013	**4.3 min**
ResNet	**0.024**	4.47 min

B. Generated captions

The generated captions of four different Automatic Image Captioning models are presented in section VIII (Appendix) below.

VI. CONCLUSIONS

By observing the experimental results, It is clear that using different CNN architectures in Image captioning models will result in different Results. So, we can reject the Null hypothesis and accept the Alternative hypothesis. The Image captioning model built using ResNet shown a comparable increase in the BLEU score compared to other CNN architectures like VGG16, MobileNet, and DenseNet. Although MobileNet and DenseNet architectures shown better results in the classification of ImageNet Dataset [11] [8] , The Residual connections in ResNet allowed smooth gradient flow which resulted in better performance of the Automatic Image captioning model. The training Time for both VGG16 and MobileNet is 4.3 min which is lower compared to DenseNet121 and ResNet50having training time of 4.7min and 4.47 minutes respectively. All the above results are obtained after Training each model for 5 epochs.

The CNN architecture that performed well in other computer vision tasks may not work better for Image captioning task. However, there are a lot of other CNN architectures left to study their effect on the performance of the Image captioning task.

VII. LIMITATIONS

we evaluated our models on Flickr8k dataset which is small in size. because of the huge computational cost and lack of better performing GPU. However, there are other commonly used benchmark datasets like Microsoft COCO Captioning and Flickr30k dataset. Also, our work is limited to using encoder-decoder architectures only. There is another type of Image captioning architectures performing better than some Encoder-Decoder architectures using attention mechanisms on images. The attention mechanism architectures use the same architectures like encoder-decoder but it passes extra information to the decoder by using attention mechanism, which is helpful in decoding the image vector into the description. These are the limitations of our work.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[2] C. Gan, T. Yang, and B. Gong, "Learning attributes equals multi-source domain generalization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 87–97, 2016.

[3] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng, "Mining semantic affordances of visual object categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4259–4267, 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[5] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, pp. 487–495, 2014.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[7] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[12] S. Bai and S. An, "A survey on automatic image caption generation," *Neurocomputing*, vol. 311, pp. 291–304, 2018.

[13] R. C. Staudemeyer and E. R. Morris, "Understanding lstm – a tutorial into long short-term memory recurrent neural networks," 2019.

[14] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.

[15] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "Treetalk: Composition and compression of trees for image descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 351–362, 2014.

[16] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 444–454, Association for Computational Linguistics, 2011.

[17] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III, "Midge: Generating image descriptions from computer vision detections," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 747–756, Association for Computational Linguistics, 2012.

[18] A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," in *Advances in neural information processing systems*, pp. 1889–1897, 2014.

[19] F. Yan and K. Mikolajczyk, "Deep correlation for matching images and text," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3441–3450, 2015.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[21] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[22] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.

[23] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in neural information processing systems*, pp. 2352–2360, 2016.

[24] F. Chollet, *Deep Learning with Python*. Manning, Nov. 2017.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318, Association for Computational Linguistics, 2002.

VIII. APPENDIX

A. VGG16 Image captioning results:

Good Captions

—Bad captions

true: a black and white dog is running through the field

pred: a black and white dog is running through the grass

BLEU: 0.8801117367933934

true: a black and white dog is running in the grass

pred: a black and white dog is running through the grass

BLEU: 0.7071067811865475

Fig. 8. Good captions for VGG16

true: a child in a pink dress is climbing up a set of stairs in an entry way

pred: a man in a red shirt and a woman in a red shirt and a woman in a white shirt and a woman in a white shirt and a woman in a white shirt

BLEU: 3.8540425273546335e-155

true: a black dog and a spotted dog are fighting

pred: a black and white dog is running through the water

BLEU: 6.848990852664275 4e-155

true: a little girl covered in paint sits in front of a painted rainbow with her hands in a bowl

pred: a man in a red shirt and a white shirt is jumping in the air

BLEU: 4.487950221566228e-155

true: a man lays on a bench while his dog sits by him

pred: a man in a red shirt and a woman in a red shirt and a woman in a white shirt and a woman in a white shirt and a woman in a blue shirt

BLEU: 3.3919880624255976e-155

true: a man in an orange hat starring at something

pred: a man in a red shirt and a white shirt is jumping in the air

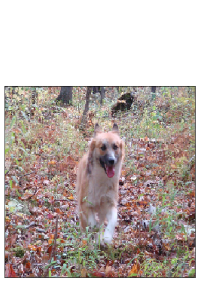
BLEU: 2.6444372049083823e-78

Fig. 9. Bad captions for VGG16

B. MobileNet Image captioning results:

Good Captions

—Bad captions



true: a brown and white dog is running through woodland

pred: a brown and white dog is running through a field

BLEU: 0.7598356856515925

Fig. 10. Good captions for MobileNet

true: a child in a pink dress is climbing up a set of stairs in an entry way

pred: a man in a blue shirt is standing on a red bench

BLEU: 4.3382562373312645e-155

true: a black dog and a spotted dog are fighting

pred: a black dog is playing in the snow

BLEU: 3.940055059819774e-78

true: a little girl covered in paint sits in front of a painted rainbow with her hands in a bowl

pred: a girl in a blue shirt is jumping into a pool

BLEU: 3.328250555827247e-155

true: a man lays on a bench while his dog sits by him

pred: a black dog is jumping over a red ball

BLEU: 9.918892480173173e-232

true: a man in an orange hat starring at something

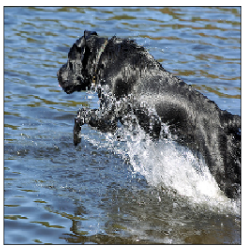
pred: a man in a blue shirt is standing on a picture

BLEU: 3.4077295621678842e-78

Fig. 11. Bad captions for MobileNet

C. DenseNet Image captioning results:

Good Captions



true: a black dog is running in the water

pred: a black dog is running in the snow

BLEU: 0.8408964152537145

true: a brown dog is running in the sand

pred: a brown dog is running in the water

BLEU: 0.8408964152537145

Fig. 12. Good captions for DenseNet

—Bad captions

true: a child in a pink dress is climbing up a set of stairs in an entry way

pred: a man in a red shirt and a white shirt and a white shirt and a man in a red shirt and a man in a red shirt and a man in a red

BLEU: 3.8540425273546335e-155

true: a black dog and a spotted dog are fighting

pred: a black dog is running through the water

BLEU: 3.940055059819774e-78

true: a little girl covered in paint sits in front of a painted rainbow with her hands in a bowl

pred: a man in a red shirt and a white shirt and a white shirt and a man in a red shirt and a man in a red shirt and a man in a red

BLEU: 4.033776337840988e-155

true: a man lays on a bench while his dog sits by him

pred: a man in a blue shirt is riding a skateboard on a skateboard

BLEU: 7.0984505763767 67e-155

true: a man in an orange hat starring at something

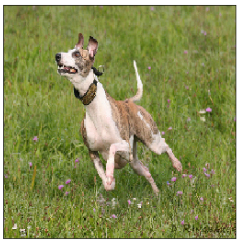
pred: a man in a red shirt and jeans is riding a skateboard

BLEU: 3.1712496890592535e-78

Fig. 13. Bad captions for DenseNet

D. ResNet Image captioning results:

Good Captions



true: a dog is running in a field

pred: a brown dog is running in a field

BLEU: 0.7071067811865475

Fig. 14. Good captions for Resnet

—Bad captions

true: a child in a pink dress is climbing up a set of stairs in an entry way

pred: a man in a red shirt is sitting on a bench

BLEU: 3.991890743256307e-155

true: a black dog and a spotted dog are fighting

pred: a black and white dog are playing with a black and white dog

BLEU: 8.16437745974496e-155

true: a little girl covered in paint sits in front of a painted rainbow with her hands in a bowl

pred: a girl in a red shirt is standing on a red and red

BLEU: 3.978230660160 86e-155

true: a man lays on a bench while his dog sits by him

pred: a man in a blue shirt is playing with a blue blanket

BLEU: 5.791739854583281e-155

true: a man in an orange hat starring at something

pred: a man in a red shirt is smiling

BLEU: 3.940055059819774e-78

Fig. 15. Bad captions for Resnet